ADAC System Design and Its Application to Mine Hunting Using SAS Imagery

Vom Fachbereich 18
Elektrotechnik und Informationstechnik
der Technischen Universität Darmstadt
zur Erlangung der Würde eines
Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Dissertation

von
Raquel Fandos, M.Sc.
geboren am 26.05.1979 in Zaragoza (Spanien)

Referent:	Prof. Dr.-Ing. Abdelhak M. Zoubir
Korreferent:	Prof. Dr.-Ing. Salim Bouzerdoum
Tag der Einreichung:	06.12.2011
Tag der mündlichen Prüfung:	27.01.2012

D 17
Darmstadt, 2012

A mis padres y hermanas

Acknowledgments

I would like to thank the people who have helped me during my doctoral study.

First, I would like to thank Prof. Dr.-Ing. Abdelhak Zoubir for his supervision. I am very grateful for the opportunity of accomplishing my PhD studies in the Signal Processing Group of the TU Darmstadt, and also for the great environment that I found there. It has been a real pleasure.

I wish to thank my co-supervisor Prof. Dr.-Ing. Salim Bouzerdoum for his hospitality during my visit to the University of Wollongong. I also want to thank Prof. Dr.-Ing. Silvia Santini and Prof. Dr.-Ing. Jutta Hanson who acted as my examiners in the PhD committee. I will not forget that Prof. em. Dr.-Ing. Eberhard Hänsler made me laugh just before my PhD presentation.

Dr. Arne Kraft and Dr. rer. nat Konstantinos Siantidis from ATLAS ELEKTRONIK GmbH always welcomed me in Bremen. Thanks to them not only for the sonar data but also for their hospitality and their time.

I would like to express my gratitude to my colleagues of the Signal Processing Group at the TU Darmstadt. Thanks to my friend and office mate Christian Debes, for his help and the good talks. It has been great to share the trips to Bremen with Stefan Leier. I also enjoyed sharing office, conference or *teilchen* with Uli Hammes, Michael Leigsnering, Feng Yin, Philipp Heidenreich, Mouhammad Alhumaidi, Nevine Demitri, Sara Al-Sayed, Sahar Khawatmi, Michael Muma, Gökhan Gul, Ahmed Mostafa, Fiky Suratman, Zhihua Lu, Waqas Sharif, Christian Weiss, Michael Fauss, Jürgen Hahn, Gebremichael Teame, Tai Fei and Ange Tchinda, as well as Renate Koschella and Hauke Fath. I also would like to thank the former PhD students and postdocs Marco Moebus, Weaam Alkhaldi and Yacine Chakhchoukh.

I wish to thank my friends Carolina, Claudia and Jithin. They contributed greatly to my happiness during the years in Darmstadt. Thanks to my friends Bea, Fabio, Julio, Pocho and Teresa for their virtual presence as well.

My parents Nemesio and Iluminada and my sisters Ana and Noelia travel with me wherever I go. Thanks to them for their love and understanding.

Thanks to Sascha for his support, his love and his laugh.

Darmstadt, 21.02.2012

Kurzfassung

Die vorliegende Dissertation behandelt Systeme zur automatischen Detektion und Klassifizierung. Es wird einerseits eine Reihe von anwendungsunabhängigen Aspekten des Entwurfs derartiger Systeme behandelt, andererseits wird die spezifische Anwendung der Unterwasserminensuche mittels der Auswertung von *Synthetic Aperture Sonar* (SAS) Bildern betrachtet.

Ein neuartiges *Resampling*-Verfahren wird vorgeschlagen, welches die Lösung zweier fundamentaler Probleme des Entwurfs von Klassifizierungssystemen erlaubt: die Auswahl des Klassifikators und die Abschätzung der optimalen Dimension der Merkmalmenge. Das Verfahren schätzt sowohl die Wahrscheinlichkeitsverteilung der Missklassifikationsrate (oder eines anderen Gütemaßes des Klassifizierungssystems) in Abhängigkeit der Dimension der Merkmalmenge als auch die Wahrscheinlichkeitsverteilung der optimalen Dimension bei Vorgabe einer Missklassifikationsrate. Letzters erlaubt insbesondere die Abschätzung von Konfidenzintervallen hinsichtlich der optimalen Dimension der Merkmalmenge. Im Gegensatz zu bereits bekannten Verfahren wird keine Annahme hinsichtlich der Verteilungsfunktion der Merkmale benötigt. Basierend auf der Wahrscheinlichkeitsverteilung des Gütemaßes wird eine Methode zur Bewertung der Qualität des Klassifikators vorgeschlagen. Das Verfahren erlaubt somit den Vergleich verschiedener Klassifikatoren ohne an eine vorgegebene Merkmalmenge gebunden zu sein. Es hebt sich hiermit von existierenden Verfahren ab.

Desweiteren wird die Bestimmung der Merkmale, welche in der optimalen Menge enthaltenen sind betrachtet. Hierzu wird eine Erweiterung des *Sequential Forward* - sowie des *Sequential Forward Floating*-Auswahlverfahren vorgeschlagen, welches deren Limitierungen abschwächt und zu besseren Ergebnissen führt.

Basierend auf den neuentwickelten Methoden wurde ein Verfahren zur automatischen Detektion und Klassifizierung von Unterwasserobjekten in SAS Bildern entwickelt. Es beinhaltet drei Schritte: Detektion, Merkmalextraktion und Klassifizierung. Zur Detektion von Objekten in Sonarbildern werden drei Segmentierungsalgorithmen verglichen: *iterative conditional modes*, *min-cut/max-flow* und *active contours*. Die Initialisierung der Segmentierungsalgorithmen hat signifikanten Einfluss auf das Ergebnis. Es werden daher neue Initialisierungsschemata, spezialisiert auf die vorliegende Anwendung, vorgeschlagen. Anschließend wird zu jedem Objekt eine umfangreiche Merkmalmenge extrahiert, welche sowohl geometrische als auch statistische Elemente beinhaltet. Diese werden derart gewählt, dass sie invariant unter Änderungen der Objektposition als auch unempfindlich gegen schlechte Segmentierungsergebnisse sind. Der

Bestimmung der optimalen Merkmalmenge mittels der erweiterten Auswahlverfahren geht die Ermittlung des besten Klassifikators anhand des vorgeschlagenen *Resampling*-Verfahrens voraus. Hierbei stehen die Klassifikatoren *k-nearest neighbor*, Mahalanobis, lineare Diskriminantenanalyse und *support vector machines* zur Auswahl.

Die vorgeschlagenen Methoden werden auf zwei Datenbanken realer SAS Bilder angewendet, welche eine Fläche von 57.000 Quadratmetern Meeresgrund mit mehr als 600 Minen unterschiedlichen Typs abbilden.

Abstract

This PhD thesis considers the problem of automatic detection and classification. On the one hand, a set of application independent design issues for classification is tackled. On the other hand, the specific application of underwater mine hunting using synthetic aperture sonar imagery is considered.

A novel resampling method is proposed in order to solve two fundamental issues involved in the design of classification systems, namely, the selection of the classifier and the estimation of the optimal feature set dimensionality. The method estimates both the probability distribution of the misclassification rate (or any other figure of merit of the classification system) subject to the size of the feature set and the probability distribution of the optimal dimensionality given a misclassification rate. The latter allows for the estimation of confidence intervals for the optimal feature set size. Unlike previous methods, no assumption for the features distribution is required. Based on the probability distribution of the figure of merit, a quality assessment for classifier performance is proposed. By contrast with previous works, the proposed algorithm allows to compare different classifiers without bonds to a specific feature set.

In addition, the problem of determining the optimal feature subset is considered. In this respect, novel extensions of the Sequential Forward Selection and Sequential Forward Floating Selection methods are proposed. It alleviates the limitations of the methods, yielding a better performance.

A system for automatic detection and classification of underwater objects using synthetic aperture sonar imagery is developed within this design framework. It consists of three steps: detection, feature extraction and classification. In order to detect the objects in the sonar images, three segmentation algorithms are compared: iterative conditional modes, min-cut/max-flow and active contours. Novel initialization schemes addressing the application at hand are proposed, since they significantly influences the final result. An extensive set of features is extracted for each object, both geometrical and statistical. They are designed to remain invariant to changes in the object position and also in poor segmentation scenarios. The selection of the optimal feature subset is accomplished by the extended feature selection algorithms, only after the resampling method has determined the best out of four classifier candidates (k-nearest neighbor, Mahalanobis', linear discriminant analysis and support vector machines).

The proposed methods have been applied to two databases of real sonar images containing over 57,000 m^2 of underwater images and 600 mines of different types.

Contents

Chapter 1

Motivation

The problem of detection and classification appears in numerous daily life situations. Children that walk along the beach looking for mollusks shells are exercising detection. When they later separate the flat colored from the striped specimens, they perform classification. An Automatic Detection And Classification (ADAC) system would do the job for the kids while they go for a swim.

ADAC has been an active field of research in the last decades [1–7]. It is used both in military and civilian applications, e.g., face recognition [8, 9], biomedical applications [10, 11], through-the-wall radar imaging [12] or mine hunting [13–15].

This thesis is divided into two parts. In the first one, some general, application independent, ADAC design issues are tackled. The second part of the thesis deals with a specific application of ADAC systems, mine hunting based on Synthetic Aperture Sonar (SAS) technology.

1.1 Publications

The following publications have been produced during the PhD tenure.

Internationally Refereed Journal Articles

- R. Fandos, S. Bouzerdoum and A. M. Zoubir. "Optimal Classification System for Speech Emotion Recognition". To be submitted to *IEEE Transactions on Audio, Speech, and Language Processing*.
- R. Fandos, A. M. Zoubir, and K. Siantidis. "Unified Design of a Feature Based ADAC System for Mine Hunting using Synthetic Aperture Sonar". Submitted to *IEEE Transactions on Geoscience and Remote Sensing*.
- R. Fandos, C. Debes, and A. M. Zoubir. "Resampling Methods for Quality Assessment of Classifier Performance and Optimal Number of Features". Submitted to *IEEE Transactions on Geoscience and Remote Sensing*.

- R. Fandos and A. M. Zoubir. "Optimal Feature Set for Automatic Detection and Classification of Underwater Objects in SAS Images". *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, pp. 454 - 468, June 2011.

Internationally Refereed Conference Papers

- R. Fandos, L. Sadamori, and A. M. Zoubir. "Sparse Representation Based Classification for Mine Hunting Using Synthetic Aperture Sonar". Accepted in the *37th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Kyoto, Japan, March 2012.

- R. Fandos, L. Sadamori, and A. M. Zoubir. "High Quality Segmentation of Synthetic Aperture Sonar Images using the Min-Cut/Max-Flow Algorithm". In *Proceedings of the 19th European Signal Processing Conference (EUSIPCO)*, vol. 1, pp. 51 - 55, Barcelona, Spain, August 2011.

- R. Fandos and A. M. Zoubir. "Enhanced Initialization Scheme for a Three-Region Markovian Segmentation Algorithm and its Application to SAS Images". In *Proceedings of the 10th European Conference on Underwater Acoustics*, vol. 3, pp. 1323 - 1331, Istanbul, Turkey, July 2010.

Part I

ADAC System Design

Chapter 2
Introduction

Given a certain scenario, ADAC systems find the objects present in the scene and assign them to one within a set of predefined classes. If the application is image based, the detection is performed by segmenting the image into the object and background regions. The classification task requires the identification of patterns within the set of detected objects. It is generally easier to design such a system when the objects are represented in some feature space that enhances those object characteristics that are meaningful for the problem at hand. While a few features might be enough in some applications, others require thousands. For the shells example, reasonable features would be measures of the object texture, such as the variance of the color intensity. The area of the shells, on the other hand, is not meaningful and may not be considered. The scheme of a typical ADAC system is shown in Fig. 2.1.

The designer of a classification system aims for a decision rule that assigns each object to a class based on its features. That is, the feature space has to be divided into as many regions as classes are supported, so an object is assigned to one or another class depending on the position value of its feature vector. Although a heuristic rule is possible, a machine learning approach provides far better results [7]: the so-called training data set, i.e., a set of objects whose corresponding classes are known beforehand (typically by hand labeling), is used to train the system, resulting in the desired decision rule. The system is then tested on a test data set, which must be different from the training set. Based on the test results, a certain figure of merit is calculated to assess the system performance, e.g., the probability of misclassification.

Several issues are to be considered in the design of an ADAC system. The first steps of the system, namely, the selection of the segmentation algorithm and the design of the features, both strongly depend on the application at hand. However, once the feature set is extracted for each object in the training data set, the design process becomes independent of the application. Therefore, the following issues are common to all ADAC system design problems:

- **Selection of the Classifier:** A broad variety of classification systems exist in the literature [7]. Each of them applies a different principle in order to divide the feature space into the class regions. For instance, Mahalanobis' classifier assigns an object to the class minimizing Mahalanobis' distance between the

Figure 2.1: General scheme of an ADAC system. If the scene is an image, the detection is typically performed by a segmentation algorithm. For each detected object, a set of features is extracted. The classifier assigns a class to each object by comparison with a training data set.

object and the different classes. The suitability of one or another classification system depends on the characteristics of the feature space.

- **Estimation of the Optimal Dimensionality of the Feature Set:** There is an exponential increase in volume of the feature space as new dimensions are added to the feature set. If the size of the training data set is finite, this implies that, as the feature space dimensionality increases, less and less observations are available per volume unit, which worsens the estimation of the features distribution. This effect is known as the curse of dimensionality [16, 17], and it is responsible for the deterioration of the system performance from a certain point as the size of the feature set increases.

- **Selection of the Actual Elements in the Feature Set:** If the amount of extracted features is higher than the optimal dimensionality of the feature set, a subset of them is to be selected. The estimation of the optimal feature subset requires the consideration of all possible feature combinations [18], which is a prohibitive task in most cases. Hence, algorithms that approximate it with a reasonable complexity are required.

2.1 State of the Art

Extensive work has been done in the fields of classification and feature selection in the last decades (see [19, 20] and references therein). Regarding the selection of the classification system, most of the existing works, e.g., [21–25], compare the performance of a collection of classifier candidates on a certain feature subset that is chosen beforehand. However, if a different feature subset were employed, the ranking might vary. Furthermore, the optimal size of the feature subset is different for different classification systems.

The estimation of the optimal dimensionality of the feature set has been an active field of research for many decades (see [2, 26] and references therein). In general, the

features distribution is assumed to be Gaussian, and the covariance matrix is assumed to be identical for all classes. Since in real applications this is often not the case, the results are unrealistic. In practice, a rule of thumb is applied: the size of the feature set should be between six and ten times smaller than the number of observations [27]. However, this is inappropriate in many cases, since the distribution of the features is completely disregarded.

A wide variety of feature selection algorithms estimating the optimal feature subset exists in the literature (see [20, 28] and references therein). In this thesis, the broadly accepted Sequential Forward Selection (SFS) and Sequential Forward Floating Selection (SFFS) methods are considered. These methods suffer from limitations, e.g., the so-called nesting effect. Thus, for example, the best subset of five features does not necessarily contain the best 4-feature subset, however, this is implicitly assumed.

2.2 Contributions

In the following, the main contributions of this thesis to the field of ADAC systems design are listed:

- **Quality Assessment of Classifier Performance:** A novel algorithm, based on resampling techniques, is developed in order to provide a quantitative measure for classifier performance. Unlike previous approaches, it avoids bonds to any specific feature subset, providing a more fair and meaningful comparison between classification systems.

- **Optimal Number of Features:** The resampling algorithm referred to above also provides confidence intervals for the optimal size of the feature subset. Unlike previous methods, it does not require assumptions on the features distribution, which results into a more accurate estimation. Knowing beforehand the expected number of elements in the optimal feature subset can drastically constraint the search space of feature selection algorithms, reducing their computational cost.

- **Feature Selection:** An extension of the SFS and the SFFS algorithms is proposed. It mitigates the limitations of the original algorithms (e.g., the nesting effect) allowing for a significant performance improvement.

2.3 Overview of Part I

The first part of this thesis consists of two main chapters. Chapter 3 describes the resampling algorithm and its application to both quality assessment of classifier performance and estimation of the optimal feature subset dimensionality. The algorithm is tested on 80 synthetic data examples and on six standard databases of real data from the UCI Machine Learning repository [29]. The problem of feature selection is considered in Chapter 4. After describing the standard SFS and SFFS methods, an extension of the algorithms, which significantly improves their performance, is proposed. The new algorithms are tested on the same six databases mentioned above. Finally, the conclusions and outlook for future work are summarized in Chapter 5.

Chapter 3

Quality Assessment of Classifier Performance and Optimal Dimensionality

When pattern recognition practitioners are required to design a classifier for a specific problem, they are generally provided with a data set of S observations. Each observation s, $1 \leq s \leq S$, has an associated feature vector $\mathbf{t} \in \mathbb{R}^N$ and a class label $c \in \{1, \ldots, C\}$, where C is the number of classes. Besides other design decisions [19], the pattern recognition expert needs to select:

1. a classification system such as k-Nearest Neighbor [7], neural networks [30], Mahalanobis' classifier [31], decision trees [32], Fisher's linear discriminant [7] or Support Vector Machines [33], among others. The classification system is responsible for setting the decision rule, namely, for dividing the feature space $\mathbb{R}^N$ into C regions. An observation is then assigned to one or another class $c \in \{1, \ldots, C\}$ depending on the position value of its feature vector.

2. an n^*-element subset of features, $\mathbf{t}^* = \{t_1^*, t_2^*, \ldots, t_{n^*}^*\}$ with $n^* \leq N$, that optimizes a certain figure of merit f for a given classification system. Typically, f corresponds to the misclassification rate, which is estimated from the available data.

Indeed, both choices are interrelated. There is no overall optimal classifier, and the superiority of one over another is application dependent. Numerous examples in the literature provide comparisons of classification systems for different applications, e.g., [21–25]. Typically, a feature set is chosen beforehand and all classifier candidates are tested on it. The classifier providing the lowest f is adopted.

The selection of a feature subset can reduce not only the cost of recognition by reducing the number of features to be collected, but it also provides a better classification accuracy due to finite sample size effects ($S < \infty$), i.e., the so-called curse of dimensionality [16]. The optimal feature subset depends on the classification system. Therefore, a subset that performs well for one classifier might provide poor results for another one, but a second subset could outperform it. In short, it is not fair to compare different classifiers with the same feature subset. However, this is normally the case. In this thesis, a novel method that overcomes this issue by assessing the classifier performance without constraints to any specific feature set is proposed.

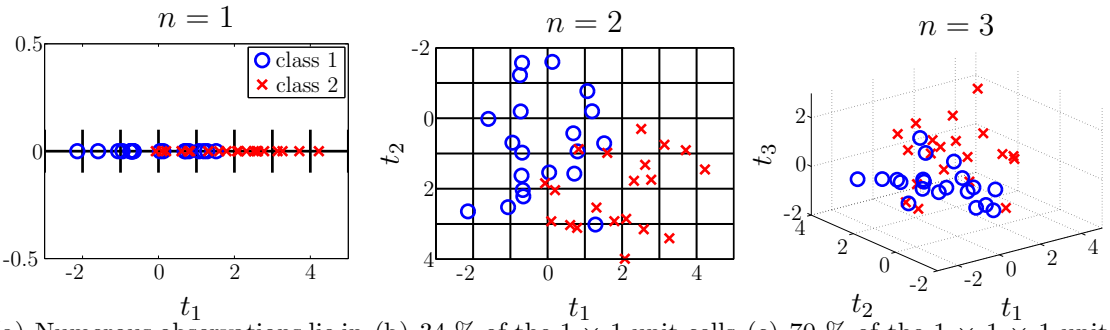
The prediction of the optimal number of features, n^*, for a given problem, has been an active field of research for several decades (see Sec. 3.5.1). Most approaches assume Gaussianity for the features and a common covariance matrix for all classes. Thus, to the best of our knowledge, no conclusive work exists. In practice, a rule of thumb suggesting that n^* should be between six and ten times smaller than S is generally applied. This rule considers neither the quality of the available features nor the possible imbalance between the classes.

Knowing n^* beforehand allows for saving computational time when a feature selection algorithm is employed in order to decide for the optimal feature subset $\mathbf{t}^*$ (see Chapter 4 for details). Furthermore, if n^* is close to N, this might indicate that the available N features do not describe the problem sufficiently, and if possible, more or better features should be extracted.

In this thesis, a novel resampling algorithm that pursues a twofold purpose is presented: on one hand, it assesses the performance of a classifier avoiding bonds to any feature subset. By doing so the best classifier out of a set of possible classifiers can be determined without restricting the procedure to a specific set of features that is suboptimal for most classifiers. On the other hand, it estimates the probability distribution of the optimal number of features n^* subject to a certain figure of merit f. This allows the prediction of the region in which the optimal number of features will be with a preset confidence. It further allows inferring confidence information on the figure of merit. Unlike previous works, no assumption for the features distribution is required.

Resampling techniques, e.g. the bootstrap, are computationally intensive tools for statistical inference in situations when either little is known about the data statistics or the available amount of data is too small to allow asymptotics based tools [34]. In the field of pattern recognition, the bootstrap has been thoroughly employed for addressing a variety of issues. A common application is the estimation of a reliable misclassification rate from a small number of observations [22, 35–39] or when analytic expressions cannot be obtained. Bootstrap techniques have also been applied for feature selection [40–42].

Neither error estimation nor feature selection are the objective of the resampling algorithm proposed in this thesis. Furthermore, there is a fundamental difference between error estimation bootstrapping and the method proposed hereafter. While the former resamples the data observations, our method resamples the features. To the best knowledge of the author, there exists no previous work where the resampling has been employed for classifier quality assessment and estimation of the optimal feature set dimensionality.



(a) Numerous observations lie in every cell of length one unit.

(b) 34 % of the 1×1 unit cells are empty.

(c) 70 % of the $1 \times 1 \times 1$ unit cells are empty.

Figure 3.1: Curse of dimensionality. Illustration of the exponential growth in volume as the feature set dimensionality increases.

The curse of dimensionality lies behind much of the work accomplished in this thesis and therefore, the chapter starts with a section devoted to its description. The fundamentals of the classification systems employed in this thesis, the k-Nearest Neighbor (k-NN), Mahalanobis' classifier, Linear Discriminant Analysis (LDA) and Support Vector Machines (SVM), are included in Sec. 3.2. Sec. 3.3 describes the resampling principle. In Sec. 3.4, resampling is employed to estimate the distribution of the figure of merit subject to the dimensionality of the feature set. Based on this, a quality assessment for classification systems is proposed. An algorithm that predicts confidence intervals for n^* is provided in Sec. 3.5.2, after a description of the state of the art in Sec. 3.5.1. In Sec. 3.6, the proposed method is applied to 80 data sets of synthetic data and in Sec. 3.7 to six standard data sets from the UCI Machine Learning repository [29]. First the performance of three classifiers are compared according to the proposed quality assessment. Subsequently, confidence intervals of the optimal dimensionality are estimated. In order to verify the effectiveness of the proposed techniques, an exhaustive search of the overall optimal feature subset is performed for the synthetic data. For the real data, two well-established feature selection techniques, the Sequential Forward Selection (SFS) and the Sequential Floating Forward Selection (SFFS), have been applied to estimate $\mathbf{t}^*$. Its dimensionality and performance of $\mathbf{t}^*$ are compared with the predicted ones.

3.1 Curse of Dimensionality

Based on the features distribution of the training database, the classification system designs a decision rule that will predict the class of any new possible observation. One could think that employing all N available features for designing the decision rule will

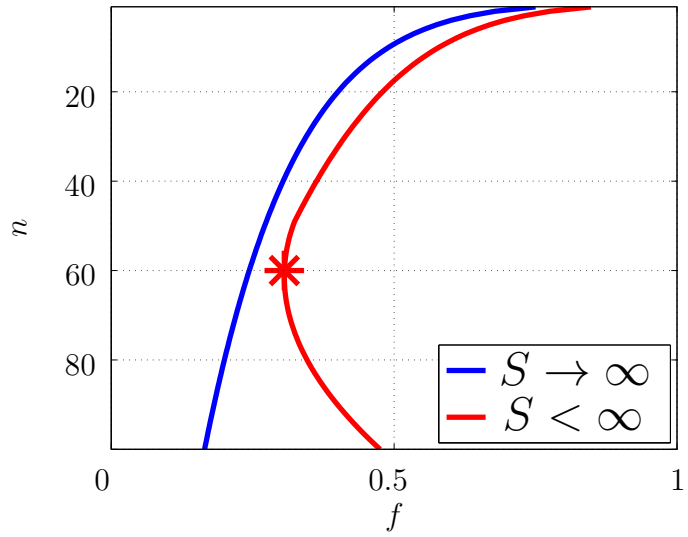


Figure 3.2: For finite number of observations, the figure of merit improves until a certain dimensionality n^* (indicated by a star) and gets worse after. If the number of observations were infinite, the performance would improve monotonically with the number of features.

provide the best performance. In this section it is shown that, due to the so-called curse of dimensionality [16, 17], it is generally advantageous to use a subset of n features, with $n < N$.

The classification rule consists of the division of the feature space $\mathbb{R}^n$ into C regions. Logically, regions of $\mathbb{R}^n$ with a high concentration of observations from a certain class c should be assigned to that very class. Consider a classification problem with two classes and 20 observations per class. A synthetic example of such a database for feature set dimensionality $n = \{1, 2, 3\}$ is illustrated in Fig. 3.1. There is an exponential increase in volume as the number of features increases, that is, less and less observations are available at each volume unit. For this reason, the effective amount of information that a fixed number of observations provides decreases as n increases. Hence, the performance, measured by the figure of merit f (e.g. the misclassification probability), improves until a certain value of n but, due to the inadequate estimation of the features distribution, gets worse for higher values of n. This effect is the curse of dimensionality and it is illustrated in Fig. 3.2. The value of n that corresponds to the best performance, i.e., the optimal number of features, is denoted by n^*. The curse of dimensionality is also known as peaking effect.

3.2 Classification Systems

In Sec. 3.4, a quality assessment for comparing classification systems is provided. The collection of classifiers employed in Sec. 3.7 for illustrating this method is described in the following. These classifiers are utilized as well in Chapter 9 for the mine hunting ADAC system as well.

Note that the focus of this thesis is not on the optimization of classification systems but rather on the choice of the best one among a set of candidates for a given application. Therefore, the classification systems are used as 'black boxes' that receive feature vectors as inputs and provide class labels as outputs. They are shortly described in the sequel for the sake of completeness.

3.2.1 k-Nearest Neighbor

The k-Nearest Neighbor (k-NN) classifier [7] is conceptually very simple. The distance between the feature vector of the observation s under consideration and the feature vectors of all training observations is measured under some norm, and the closest k training observations, the nearest neighbors, are selected. The observation s is assigned to the class c to which most of its k nearest neighbors belong. To avoid draws, k is normally chosen to be odd. The results presented in this thesis are calculated with $k = 5$. The Euclidean norm has been used.

3.2.2 Mahalanobis' Classifier

Mahalanobis' classifier [31] assumes that the feature set $\mathbf{t} \in \mathbb{R}^n$ of the observations belonging to class c follows a multivariate Gaussian distribution. Thus, its probability density function (pdf) reads,

$$p_c(\mathbf{t}) = \frac{1}{(2\pi)^{\frac{n}{2}}} |\boldsymbol{\Sigma}_c|^{-\frac{1}{2}} \cdot \exp\big[-\frac{1}{2}(\mathbf{t}-\boldsymbol{\mu}_c)'\boldsymbol{\Sigma}_c^{-1}(\mathbf{t}-\boldsymbol{\mu}_c)\big], \qquad c \in \{1,\ldots,C\}, \tag{3.1}$$

where $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$ are the mean and covariance matrix corresponding to class c, respectively. They are approximated by the sample mean, $\hat{\boldsymbol{\mu}}_c$, and sample covariance matrix, $\hat{\boldsymbol{\Sigma}}_c$, which are estimated from the training data.

The classification decision is based on the likelihood ratio,

$$\frac{p_{c_1}(\mathbf{t})}{p_{c_2}(\mathbf{t})} = \frac{|\hat{\boldsymbol{\Sigma}}_{c_2}|^{\frac{1}{2}}}{|\hat{\boldsymbol{\Sigma}}_{c_1}|^{\frac{1}{2}}} \cdot \exp\big\{-\frac{1}{2}\big[(\mathbf{t}-\hat{\boldsymbol{\mu}}_{c_1})'\hat{\boldsymbol{\Sigma}}_{c_1}^{-1}(\mathbf{t}-\hat{\boldsymbol{\mu}}_{c_1}) - (\mathbf{t}-\hat{\boldsymbol{\mu}}_{c_2})'\hat{\boldsymbol{\Sigma}}_{c_2}^{-1}(\mathbf{t}-\hat{\boldsymbol{\mu}}_{c_2})\big]\big\}. \tag{3.2}$$

Taking the logarithm, Eq. (3.2) can be rearranged,

$$\ln \frac{p_{c_1}(\mathbf{t})}{p_{c_2}(\mathbf{t})} = -\frac{1}{2}\big[\ln|\hat{\mathbf{\Sigma}}_{c_1}| + (\mathbf{t}-\hat{\boldsymbol{\mu}}_{c_1})'\hat{\mathbf{\Sigma}}_{c_1}^{-1}(\mathbf{t}-\hat{\boldsymbol{\mu}}_{c_1})\big] + \frac{1}{2}\big[\ln|\hat{\mathbf{\Sigma}}_{c_2}| + (\mathbf{t}-\hat{\boldsymbol{\mu}}_{c_2})'\hat{\mathbf{\Sigma}}_{c_2}^{-1}(\mathbf{t}-\hat{\boldsymbol{\mu}}_{c_2})\big]. \tag{3.3}$$

Note that the first and second terms depend exclusively on classes c_1 and c_2, respectively. Each term has two contributions: the logarithm of the covariance matrix determinant and Mahalanobis' distance between the feature set and the distribution of the corresponding class. Associating a constant J_c to each class $c \in \{1,\ldots,C\}$, Mahalanobis' classifier assigns an observation with feature vector $\mathbf{t}$ to class c_1 if

$$\ln \frac{p_{c_1}(\mathbf{t})}{p_{c_2}(\mathbf{t})} \geq J_{c_1} - J_{c_2} \quad \forall c_2 \in \{1,\ldots,C\}, c_2 \neq c_1. \tag{3.4}$$

The constants J_c are chosen taking into account the characteristics of the distributions at hand and the design objective. For example, if all classes are equiprobable and the design objective is the minimization of the overall misclassification probability, then $J_c = 0\ \forall c$, and observations will be assigned to the class maximizing the probability distribution. By contrast, if the focus is on the minimization of the misclassification probability of a certain class c_1 (for instance, in a detection problem), then J_{c_1} should be smaller than $J_{c_2}\ \forall c_2 \in \{1,\ldots,C\}, c_2 \neq c_1$, prioritizing the correct classification of c_1 observations at the expenses of misclassifying more c_2 observations, $c_2 \neq c_1$. In Fig. 3.3, an example illustrates Mahalanobis' classifier for a feature set of dimension $n = 1$.

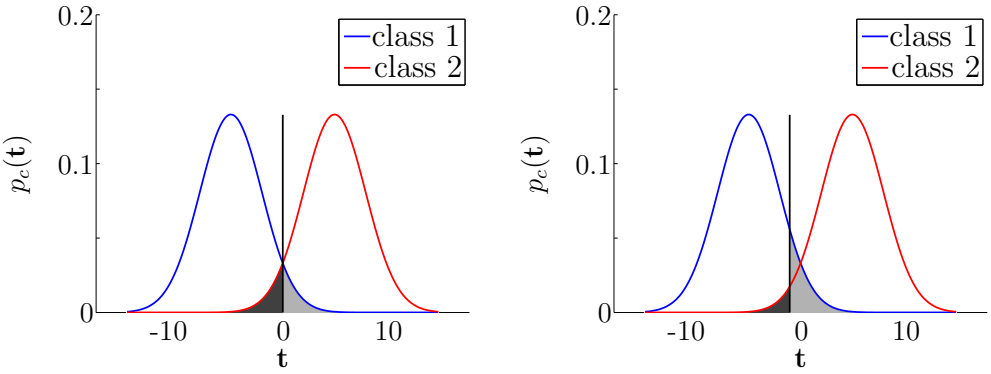
3.2.3 Linear Discriminant Analysis

The Linear Discriminant Analysis (LDA) classifier [7] assumes that the covariance matrix in Eq. (3.1) is identical for all classes. Although this appears as a loss of generality with respect to Mahalanobis' classifier, it might be an advantage when the number of training observations is reduced. Due to the curse of dimensionality (see Sec. 3.1), the estimation of the sample covariance matrix for the different classes is prone to be inaccurate, and it can be shown that the pooled covariance matrix,

$$\hat{\mathbf{\Sigma}} = \frac{1}{C}\sum_{c=1}^{C}\hat{\mathbf{\Sigma}}_c. \tag{3.5}$$

constitutes a better estimation [43]. In this case, Eq. (3.3) simplifies,

$$\ln \frac{p_{c_1}(\mathbf{t})}{p_{c_2}(\mathbf{t})} = (\hat{\boldsymbol{\mu}}_{c_1} - \hat{\boldsymbol{\mu}}_{c_2})'\hat{\mathbf{\Sigma}}^{-1}\mathbf{t} - \frac{1}{2}(\hat{\boldsymbol{\mu}}_{c_1} + \hat{\boldsymbol{\mu}}_{c_2})'\hat{\mathbf{\Sigma}}^{-1}(\hat{\boldsymbol{\mu}}_{c_1} - \hat{\boldsymbol{\mu}}_{c_2}), \tag{3.6}$$



(a) If both classes are equiprobable, the overall misclassification probability is minimized when the decision threshold is at the position where both pdfs intersect.

(b) If the correct classification of class 2 observations is prioritary, the threshold is shifted to the left. As a result, the misclassification of class 1 observations increases.

Figure 3.3: Mahalanobis' classifier for a two class problem and dimensionality $n = 1$. The pdf p_c of the feature set $\mathbf{t}$ is depicted for $c = \{1, 2\}$. A vertical line indicates the decision threshold: observations to its left and right are assigned to class 1 and 2, respectively. The areas in gray indicate the probability of misclassification for both classes.

which is a linear discriminant function of $\mathbf{t}$. Hence, Eq. (3.6) defines an hyperplane $\mathbf{w} \cdot \mathbf{t} - w_0 = 0$, where $\mathbf{w} = (\hat{\boldsymbol{\mu}}_{c_1} - \hat{\boldsymbol{\mu}}_{c_2})'\hat{\boldsymbol{\Sigma}}^{-1}$ and $w_0 = \frac{1}{2}(\hat{\boldsymbol{\mu}}_{c_1} + \hat{\boldsymbol{\mu}}_{c_2})'\hat{\boldsymbol{\Sigma}}^{-1}(\hat{\boldsymbol{\mu}}_{c_1} - \hat{\boldsymbol{\mu}}_{c_2})$. The decision rule is established by comparing Eq. (3.6) with $J_{c_1} - J_{c_2}$, $\forall c_1, c_2 \in \{1, \ldots, C\}$, as indicated by Eq. (3.4).

The LDA classifier is used in Chapter 9 for the mine hunting ADAC system due to the poor performance provided by Mahalanobis' classifier in that case.

3.2.4 Support Vector Machines

The main limitation of Mahalanobis' and LDA classifiers is the Gaussianity assumption. Features are indeed rarely Gaussian and hence, the performance of the classifiers degrades. Support Vector Machines (SVM) constitute a powerful tool able to deal with such scenarios. SVM do not require the estimation of the features distribution and, rather than minimizing the misclassification rate, they focus on the maximization of the decision confidence. First in this section, the linear SVM [33, 44] are presented. Subsequently, an extension that allows for non-linear classification is described.

Like the LDA classifier, linear SVM use hyperplanes to divide the feature space into C regions. In the following, $C = 2$ is assumed and a single hyperplane, $\mathbf{w} \cdot \mathbf{t} - w_0 = 0$, is required. At the end of the section an extension for $C > 2$ is provided.

The training database is employed in order to define the hyperplane. Ideally, observations from different classes should stay at different sides. This can be mathematically formulated as follows. Each observation s with feature vector $\mathbf{t}_s$ and class $c \in \{1, 2\}$, has an associated value $\kappa_s \in \{-1, 1\}$. For class 1 observations $\kappa_s = -1$, and κ_s equals 1 for observations belonging to class 2.

The hyperplane defines two half-spaces of observations classified with large confidence:

$$\mathbf{w} \cdot \mathbf{t} - w_0 \geq 1 \tag{3.7}$$

$$\mathbf{w} \cdot \mathbf{t} - w_0 \leq -1 \tag{3.8}$$

The distance between these two half-spaces is referred to as margin and equals $\frac{2}{||\mathbf{w}||}$. Two conditions define $\mathbf{w}$ and w_0. On the one hand, the margin should be as large as possible:

$$\min ||\mathbf{w}||. \tag{3.9}$$

On the other hand, all training observations should be correctly classified with large confidence,

$$\kappa_s(\mathbf{w} \cdot \mathbf{t}_s - w_0) \geq 1, \quad 1 \leq s \leq S. \tag{3.10}$$

Fig. 3.4 illustrates this with an example. The feature set dimensionality is $n = 2$ and therefore, the hyperplane reduces to a straight line. Clearly, there is a compromise between Eqs. (3.9) and (3.10) and often, the second condition is unfulfilled for a certain amount of observations in order to increase the margin. This can be expressed as

$$z(\mathbf{t}_s, \mathbf{w}, w_0, \kappa_s) = \max(0, 1 - \kappa_s(\mathbf{w} \cdot \mathbf{t}_s - w_0)). \tag{3.11}$$

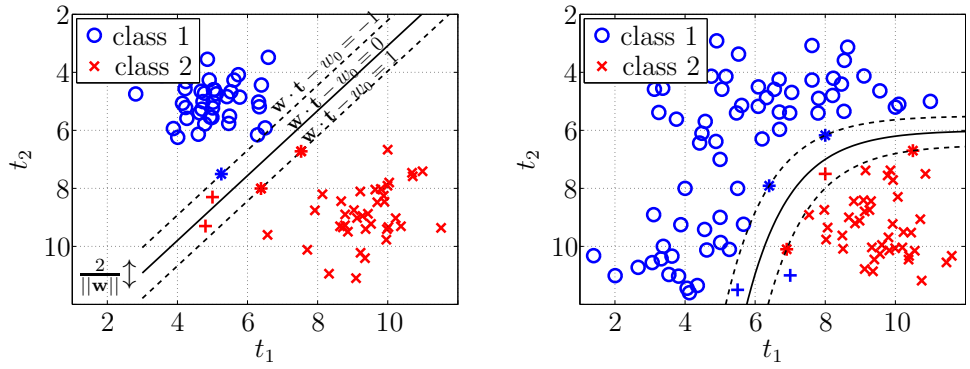
For the observations classified with a large confidence, $z(\mathbf{t}_s, \mathbf{w}, w_0, \kappa_s) = 0$. By contrast, $z(\mathbf{t}_s, \mathbf{w}, w_0, \kappa_s)$ increases with the distance between $\mathbf{t}_s$ and $\kappa_s(\mathbf{w} \cdot \mathbf{t}_s - w_0) = 1$ for those observations such that $\kappa_s(\mathbf{w} \cdot \mathbf{t}_s - w_0) < 1$. Hence, the combination of a large margin with the correct classification of most training observations reads

$$\min_{\mathbf{w}, w_0} \Big\{ \frac{1}{2}||\mathbf{w}||^2 + M \sum_{s=1}^{S} z(\mathbf{t}_s, \mathbf{w}, w_0, \kappa_s) \Big\}, \tag{3.12}$$

where M is a parameter that controls the relative importance of both requirements. Eq. (3.12) can be solved employing Lagrange multipliers (for details, see [45]). It is advantageous to express this optimization problem in its dual form,

$$\max_{\upsilon_s} \Big\{ \sum_{s=1}^{S} \upsilon_s - \frac{1}{2} \sum_{s,s'} \upsilon_s \upsilon_{s'} \kappa_s \kappa_{s'} (\mathbf{t}_s \cdot \mathbf{t}_{s'}) \Big\}, \tag{3.13}$$

subject to $0 \leq \upsilon_s \leq M$ and $\sum_{s=1}^{S} \upsilon_s \kappa_s = 0$, where υ_s are the Lagrange multipliers. It can be demonstrated that $\upsilon_s \neq 0$ only for the observations s fulfilling $\kappa_s(\mathbf{w} \cdot \mathbf{t}_s - w_0) = 1$,



(a) Linear SVM. (b) Non-linear SVM.

Figure 3.4: SVM. The support vectors are indicated by stars. They define a hyperplane that separates the regions corresponding to classes 1 and 2. For the non-linear SVM, the hyperplane is defined in the kernel domain (radial basis function). In the feature space, the border between regions is not linear. The observations that are not classified with a large confidence (they do not accomplish Eq. (3.10)) are indicated with a plus sign.

the so-called support vectors. Hence, the values of $\mathbf{w}$ and w_0 are determined exclusively by the support vectors (see Fig. 3.4). The solution to this problem is obtained by quadratic programming techniques, and it reads:

$$\mathbf{w} = \sum_{s=1}^{S} \upsilon_s \kappa_s \mathbf{t}_s \tag{3.14}$$

$$w_0 = \mathbf{w} \cdot \mathbf{t}_s - \kappa_s, \tag{3.15}$$

where w_0 is computed from any support vector.

Note that the dot product between pairs of feature sets in Eq. (3.13) can be substituted by a more general kernel function,

$$\max_{\upsilon_s} \Big\{ \sum_{s=1}^{S} \upsilon_s - \frac{1}{2} \sum_{s,s'} \upsilon_s \upsilon_{s'} \kappa_s \kappa_{s'} \Phi(\mathbf{t}_s, \mathbf{t}_{s'}) \Big\}. \tag{3.16}$$

By doing so, the SVM algorithm defines the hyperplane into the kernel space. Back to the feature space, the border between class regions is not linear anymore. This constitutes the main strength of SVM, since it allows for separating classes of arbitrary distributions. The radial basis function,

$$\Phi(\mathbf{t}_s, \mathbf{t}_{s'}) = \exp(-d||\mathbf{t}_s - \mathbf{t}_{s'}||^2), \quad d > 0, \tag{3.17}$$

is often employed as kernel function, and it has also been adopted in this thesis.

There are two parameters that need to be estimated for a non-linear SVM with radial basis kernel, d and M. The SVM implementation employed in this thesis [46] finds

them by means of a grid search, which determines the combination of values performing best for the given data.

Furthermore, the SVM in [46] allow for $C > 2$. A binary problem is built for each class: a hyperplane separates that class from the others, which are considered as a single class. Observations are then classified according to all hyperplanes. Finally, they are assigned to the class that has been chosen with the largest confidence.

3.3 Resampling Techniques

The resampling technique proposed in this thesis is similar to the bootstrap. In this section, the standard bootstrap is described. Subsequently, the differences between it and the resampling method employed thereafter are highlighted.

The bootstrap is a technique that allows for statistical inference of parameters when few data samples are at hand or too little is known about the statistics of the problem. Despite being computationally demanding, it has gained importance in the last years, as the available computer power is exponentially increasing [47].

Let $\mathbf{z} = \{z_1, z_2, \ldots, z_N\}$ be a set of measurements, which are realizations of a random variable set $\mathbf{Z} = \{Z_1, Z_2, \ldots, Z_N\}$, drawn from a distribution $p_{\mathbf{Z}}$. Typically, one is interested in the distribution of some parameter estimator $\hat{\theta} = \hat{\theta}(\mathbf{Z})$. For example for the mean $\mu_{\mathbf{Z}}$, we could be interested in the distribution of the sample mean: $\hat{\theta} = \hat{\mu}_{\mathbf{Z}} = \frac{1}{N}\sum_{j=1}^{N} Z_j$. If $p_{\mathbf{Z}}$ is known, it is possible to exactly evaluate the distribution of the parameter estimator $\hat{\theta}$, $p_{\hat{\theta}}$. However, if $p_{\mathbf{Z}}$ is unknown or $\hat{\theta}$ is some complicated estimator, its distribution cannot be derived in a closed form. Provided that enough data is available, asymptotic arguments could be used and the distribution of $\hat{\theta}$ could be approximated. If this is not the case, we may apply the bootstrap.

The bootstrap paradigm dictates that the unknown distribution $p_{\mathbf{Z}}$ is approximated by the empirical distribution of the data $\hat{p}_{\mathbf{Z}}$. Hence, N_{B} bootstrap samples $\mathbf{z}'_b = \{z'_1, z'_2, \ldots, z'_n\}$, $1 \leq b \leq N_{\mathrm{B}}$, are generated from $\mathbf{z}$ by drawing at random with replacement. For each sample $\mathbf{z}'_b$ a bootstrap parameter estimate, $\hat{\theta}'_b = \hat{\theta}(\mathbf{z}'_b)$, is obtained. Thus, the distribution of $\hat{\theta}$, $p_{\hat{\theta}}$, is approximated by the distribution of $\hat{\theta}'$, $p_{\hat{\theta}'}$, provided a large number N_{B} of bootstrap parameter estimates. As a rule of thumb, $N_{\mathrm{B}} > 50$ for variance estimation and $N_{\mathrm{B}} > 1000$ for confidence intervals estimation is suggested [48].

The standard bootstrap employs replacement and n equals the total number of measurements N. However, other schemes are also possible. The jackknife for example

resamples the data by systematically deleting a fixed number of elements from the data set $\mathbf{z}$ [49]. The subsampling bootstrap on the other hand chooses a block size n, $n < N$, and obtains each bootstrap sample by selecting n consecutive elements from $\mathbf{z}$ [50].

Like the standard bootstrap, the application presented in this thesis performs sampling at random, and the elements do not need to be consecutive. However, no replacement is employed and $n < N$ (see Sec. 3.4). Furthermore, in the context of classification, the bootstrap is generally applied to resample observations (e.g. for error estimation). By contrast, the application proposed in this thesis resamples the feature set. Note that the features are considered random conditional on the data.

3.4 Quality Assessment of Classifier Performance

The natural figure of merit f of a classifier is the overall misclassification probability:

$$P_{\mathrm{m}} = \sum_{c_1} P(c_1) \cdot \sum_{c_2 \neq c_1} P(c_2|c_1), \quad c_1, c_2 \in \{1, \ldots, C\}, \tag{3.18}$$

where $P(c_1)$ is the prior probability of class c_1 and $P(c_2|c_1)$ is the probability of deciding for class c_2 when the actual class is c_1. In some applications though, other measures might be of interest. For instance, in a detection scenario one should minimize the error rate for one class (missed detection rate) while keeping the error rate of the other class (false alarm rate) under a certain threshold. In Chapter 9, a novel figure of merit is developed. It takes into account the specific characteristics of mine hunting databases where more than one kind of mines as well as a considerable amount of clutter are present.

Independently of its metric, f is usually estimated in the following manner. First, a feature subset is selected. The S data observations are divided into the training and the test set so that the classifier is designed according to the former and it is tested on the latter. There are several strategies on how to accomplish this division [19]. The re-substitution (all available data are employed as both training and test sets) is the computationally most efficient one but it is optimistically biased. The leave-one-out technique trains the system on all available observations except one, and then tests it on the remaining observation. This procedure is repeated for all available observations. The leave-one-out technique fully exploits the available data and is unbiased, but it is computationally expensive. A good trade-off is, for instance, the 5-fold cross validation approach (which is employed for the mine hunting ADAC system design presented in

Chapter 9). The S available observations are randomly divided into five groups. In turns, the system is designed according to four of them and tested on the fifth one. For all of these methods, a single f value is eventually obtained.

However, this value of f is influenced by the finite number of observations and also by the pre-selected feature subset. The former influence decreases as the size of S increases. If S is too small, different techniques (among them the bootstrap) can be employed in order to obtain a better estimation of f [35, 36]. The pre-selection of a feature subset undermines the comparison of classification systems. A solution to the latter problem is proposed in the following.

Consider f to be a random variable conditional on the number of features n. Define a set of possible sizes, $n_i \in \{n_1, n_2, \ldots, n_{\max}\}$, $n_{\max} \leq N$, and apply resampling techniques to estimate the empirical conditional probability density function, $\hat{p}_{f|n_i}$. After choosing the number of samples N_{B}, repeat the following steps for each n_i:

- **Step 1.** Select randomly from $\mathbf{t} = \{t_1, t_2, \ldots, t_N\}$ a set of n_i features to obtain a sample $\mathbf{t}'_b = \{t'_1, t'_2, \ldots, t'_{n_i}\}$
- **Step 2.** Compute the figure of merit estimate $f'_{b,n_i} = f(\mathbf{t}'_b)$
- **Step 3.** Repeat Steps 1 and 2 N_{B} times to obtain a set of figure of merit estimates, $\{f'_{1,n_i}, \ldots, f'_{N_{\mathrm{B}},n_i}\}$, from which the empirical distribution $\hat{p}_{f|n_i}$ can be derived

No replacement is applied, that is, a sample $\mathbf{t}'_b$ contains each feature t_j at most once. Otherwise, some classification systems would fail, e.g., Mahalanobis' classifier, which requires the inverse of the covariance matrix.

The estimation of $\hat{p}_{f|n_i}$ from $\{f'_{1,n_i}, \ldots, f'_{N_{\mathrm{B}},n_i}\}$ may be done using either a parametric model (e.g., a Gaussian or Gamma distribution) or a non-parametric approach, such as kernel probability density estimation or histogram techniques [51].

The representation of $\hat{p}_{f|n_i}$ in the $f - n$ plane (see examples in Secs. 3.6.1 and 3.7.1) illustrates the performance of the classifier. The comparison of $\hat{p}_{f|n_i}$ for different classifier candidates allows for selecting the most suitable one for the problem at hand. There exists different options for quantifying the performance of a classifier from $\hat{p}_{f|n_i}$, the simplest one being the selection of the classifier providing the lowest figure of merit estimate,

$$f_b^{\min} = \min\left\{f'_{b,n_i}, b \in \{1, \ldots, N_{\mathrm{B}}\}, n_i \in \{n_1, \ldots, n_{\max}\}\right\}. \tag{3.19}$$

Indeed, at this stage, the feature subset associated to $f_b^{\min}$ may be selected as optimal feature subset and the corresponding classifier as optimal classification system. However, a feature selection algorithm (see Chapter 4) is likely to significantly improve this result (see examples in Secs. 3.6 and 3.7, respectively). The more the energy of $\hat{p}_{f|n_i}$ concentrates towards low values of f, the higher is the probability that the feature selection algorithm reaches a considerably lower figure of merit. For this reason, rather than in $f_b^{\min}$, we are interested in the classifier candidate whose $\hat{p}_{f|n_i}$ energy concentrates in lower f values. Hence, we propose to select the classifier maximizing the following expression:

$$Q = \frac{1}{n_{\max} - n_1} \sum_{n_i=n_1}^{n_{\max}} \Delta n_i \cdot \int_0^1 w(f) \cdot \hat{p}_{f|n_i} \mathrm{d}f, \tag{3.20}$$

where $\Delta n_i = n_{i+1} - n_i$ and $w(f)$ must be a monotonically decreasing function, e.g., $w(f) = f^{-\psi}$ with $\psi > 0$. This ensures that the contribution of $\hat{p}_{f|n_i}$ to Q is greater for smaller f values. The value of ψ is to be selected by the designer. The higher its value, the more weight is assigned to lower f. If $\psi \to \infty$, the rule reduces to choosing the classifier with the lowest figure of merit estimate. The simulation study in Sec. 3.6 concludes that a good value of ψ is $\psi = 1$. Note that for $\psi = 0$, Eq. (3.20) equals 1 independently of the classifier performance.

This quality assessment might be unsuitable in some situations. For example, consider a classification problem with a strong effect of the curse of dimensionality. In such case, the contribution of $\hat{p}_{f|n_i}$ to Q will be small for most n_i, probably resulting in a rather low Q value. However, $\hat{p}_{f|n_i}$ could reach very low f values for the optimal number of features. If compared with another classifier that does not suffer from the curse of dimensionality, the former classifier might be unfairly discarded. In such cases, it is more convenient to define Q as:

$$Q = \max\Big\{ \int_0^1 w(f) \cdot \hat{p}_{f|n_i} \mathrm{d}f, \quad n_i \in \{n_1, \ldots, n_{\max}\}\Big\}. \tag{3.21}$$

Note that, since the samples $\mathbf{t}'_b$ are obtained from the set of available features, $\mathbf{t} = \{t_1, t_2, \ldots, t_N\}$ influences Q. However, the quality assessment is not bound to any specific subset $\mathbf{t}^* = \{t_1^*, t_2^*, \ldots, t_{n^*}^*\}$, $n^* < N$. Hence, given a data set and a feature space $\mathbf{t} \in \mathbb{R}^N$, the performance of a set of classifier candidates is first assessed, and the best one is chosen. Then, a feature selection algorithm is used only on that classifier in order to find the optimal subset $\mathbf{t}^*$.

The computational cost of the method is regarded in Sec. 9.5, for the mine hunting ADAC examples.

3.5 Optimal Number of Features

3.5.1 State of the Art

The estimation of the optimal number of features n^* was subject of thorough research mainly in the 1970s and 1980s. In the following, a summary of the main works is provided.

In [17], the curse of dimensionality is regarded and average performances over all possible problems generated by certain *a priori* distributions are provided.

Naturally, the performance of a feature set $\mathbf{t}$ depends on the so-called class separability, that is, on how distinct and compact the n-dimensional regions occupied by the feature vectors of the different classes are. In [52] the class separability is measured by Mahalanobis' distance. As long as it increases proportionally to the dimensionality, no performance loss is experienced.

Another review paper [26] summarizes the related works published before 1982. Most of them [53–55] assume equiprobable classes, multivariate Gaussian distributions with a common covariance matrix for all classes. So does [56], which proposes that the number of observations S should increase linearly with n when linear discriminants are used, and quadratically in the case of quadratic discriminants.

The class unbalance constitutes a further issue. Agreed that equiprobable classes result in optimal performance [52,55], solutions are proposed assuming again Gaussianity and a common covariance matrix.

The following rule of thumb has been proposed [27]: n^* should be between six and ten times smaller than S, as long as the class separability, measured by Bhattacharyya's distance [57], increases with the dimensionality. Equiprobable classes are assumed.

It is noted that previous work in this area assumes knowledge of the distribution of the data or the prior probabilities of the classes. Although bootstrapping techniques are often used in the context of classification for other applications (error estimation [22,35–39] , feature selection [40–42], random forests [58]), to the best of our knowledge, no attempt has been made to employ resampling to overcome these limitations and predict the optimal number of features. A prediction of the optimal number of features allows to strongly reduce computation time for the feature selection algorithm applied subsequently.

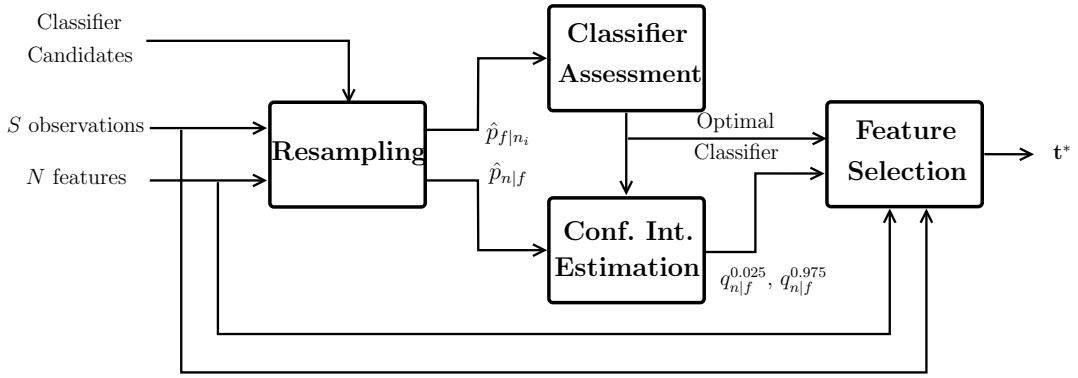


Figure 3.5: Bock diagram of the design process for a classification system as proposed in this thesis. The first three blocks are described in this chapter, while feature selection is considered in Chapter 4.

3.5.2 The Resampling Approach

A new approach to the problem, which employs the resampling algorithm described in Sec. 3.4, is proposed in the sequel. Unlike the methods described above, no assumption about the distribution of the data or the prior probabilities of the classes is required (see simulations in Sec. 3.6).

Bayes' theorem states

$$p_{n|f} = \frac{p_n \cdot p_{f|n}}{p_f}. \tag{3.22}$$

In the following we assume p_n as well as p_f to be uniformly distributed. Practically this means that given a classification problem at hand, all numbers of features $n = \{1, \ldots, N\}$ and all metrics in an interval $[f_{\min}, f_{\max}]$ are equally likely. This is the natural assumption drawn when no *a priori* knowledge is available. Furthermore, the results in Sec. 3.6 support its validity. It is noted, however, that any knowledge on p_n or p_f can easily be incorporated in Eq. (3.22) and all following conclusions still hold. Hence, let us express Eq. (3.22) as:

$$p_{n|f} = A \cdot p_{f|n}, \tag{3.23}$$

where the constant A assures that $\sum_n p_{n|f} = 1$. An empirical estimation of $p_{f|n}$, $\hat{p}_{f|n_i}$, is provided by the resampling algorithm proposed in Sec. 3.4 and hence, $\hat{p}_{n|f} = A \cdot \hat{p}_{f|n_i}$, with $n_i = n_1, \ldots, n_{\max}$.

It is known [28] that if the feature elements of $\mathbf{t}^*$ are carefully chosen, the figure of merit can significantly be improved. That is, if instead of selecting the n^* features in $\mathbf{t}^*$ in a random manner (as we have done for obtaining $\{f'_{b,n_i}\}$) we apply a feature

selection algorithm, we will very likely reach a figure of merit f^* that is smaller than $f_b^{\min}$. Therefore, unlike for the classifier quality assessment application, $\hat{p}_{f|n_i}$ cannot be assigned to the histogram of f'_{b,n_i}, $H_{f|n_i}$, because $H_{f|n_i} = 0$ for $f < f_b^{\min}$, $n_i = n_1, \ldots, n_{\max}$, and that is precisely the region where f^* is expected. Hence, a model that extrapolates $\hat{p}_{f|n_i}$ for $f < f_b^{\min}$ is required. A thorough study of the statistics of f'_{b,n_i} has been accomplished for several synthetic and real data examples (see Secs. 3.6 and 3.7). It has been observed that the probability distribution of $\{f'_{b,n_i}\}$ is reasonably close to the Gaussian, which has been chosen to model $\hat{p}_{f|n_i}$. Thus, the mean and covariance matrix are approximated by the sample mean and sample covariance matrix calculated from the figure of merit estimates, $\{f'_{1,n_i}, \ldots, f'_{N_\mathrm{B},n_i}\}$. Note that while the traditional methods referred in Sec. 3.5.1 assume Gaussianity for the distribution of the features subject to the class, $p_{\mathbf{t}|c}$, $c = \{1, \ldots, C\}$, we assume the figure of merit conditional to the number of features, $\hat{p}_{f|n_i}$, to be Gaussian. Among all simulated and real data sets, only one real data set significantly diverges from this model, resulting in a poor confidence interval estimation. To overcome this issue, a non-parametric approach based on kernel density estimation is proposed (see Sec. 3.7.3).

As referred above, feature selection algorithms are expected to find a feature subset outperforming $f_b^{\min}$ (see Eq. (3.19)). Indeed, if the figure of merit associated with the feature subset provided by the feature selection method is not lower than $f_b^{\min}$, then we should adopt the feature subset producing $f_b^{\min}$ as optimal feature subset. Therefore, we limit our study to $f \leq f_b^{\min}$. The 95 % confidence interval of $\hat{p}_{n|f}$, $q_{n|f}^{0.025} \leq n \leq q_{n|f}^{0.975}$ subject to $f \leq f_b^{\min}$ delimits the region where $\{f^*, n^*\}$ is most probably expected. Thus, more than a precise prediction of the optimal number of features, our algorithm provides a region in the $f - n$ space where it is most likely located. In Sec. 3.6, where different examples with simulated data are studied and the overall optimal feature subset is found by means of exhaustive search, 25 % confidence intervals are employed in order to study the convergence of the confidence intervals to their real value.

The size of the confidence interval $q_{n|f}^{0.025} \leq n \leq q_{n|f}^{0.975}$ naturally depends on the value of f. Thus, when a feature selection algorithm is applied (see Chapter 4), the n search space can be restricted beforehand to the interval $n_\mathrm{m} \leq n \leq n_\mathrm{M}$, with

$$n_\mathrm{m} := \min\{q_{n|f}^{0.025}\} \quad \forall f \leq f_b^{\min}, \tag{3.24}$$

$$n_\mathrm{M} := \max\{q_{n|f}^{0.975}\} \quad \forall f \leq f_b^{\min}. \tag{3.25}$$

Taking into account that the size of $q_{n|f}^{0.975} - q_{n|f}^{0.025}$ decreases as f decreases, n_m and n_M can be adapted in the following manner at each iteration l of an iterative feature

selection process:

$$\begin{aligned} \text{if} \quad & f_l < \min\{f_1, \ldots, f_{l-1}\}, \quad \text{then} \\ & n_{\mathrm{m}} := q_{n|f_l}^{0.025} \\ & n_{\mathrm{M}} := q_{n|f_l}^{0.975}, \end{aligned} \tag{3.26}$$

where f_l is the value of the figure of merit at iteration l, calculated from the estimated optimal feature subset at that iteration, $f_l = f(\mathbf{t}_l^*)$.

Finally, a block diagram of the complete design process for classification systems proposed in this thesis is depicted in Fig. 3.5. Given a set of S observations with feature set $\mathbf{t}$, and once a set of classifier candidates has been pre-selected, we proceed:

- **Step 1.** For each classifier candidate, apply resampling. For $n_i = n_1, \ldots, n_{\max}$ and for $b = 1, \ldots, N_{\mathrm{B}}$:
  a) Select from $\mathbf{t} = \{t_1, t_2, \ldots, t_N\}$ a random set of n_i features, $\mathbf{t}'_b = \{t'_1, t'_2, \ldots, t'_{n_i}\}$
  b) Compute the figure of merit $f'_{b,n_i} = f(\mathbf{t}'_b)$
- **Step 2.** Assess the quality of each classifier:
  a) From the set $\{f'_{1,n_i}, \ldots, f'_{N_{\mathrm{B}},n_i}\}$ estimate $\hat{p}_{f|n_i}$, $n_i \in \{n_1, n_2, \ldots, n_{\max}\}$
  b) Calculate Q for each classifier, choose the one with the highest Q
- **Step 3.** For the selected classifier, find n^*:
  a) From $\{f'_{1,n_i}, \ldots, f'_{N_{\mathrm{B}},n_i}\}$ estimate $\hat{p}_{f|n_i}$, $n_i = n_1, \ldots, n_{\max}$ by fitting to a Gaussian distribution
  b) Calculate $\hat{p}_{n|f} = A \cdot \hat{p}_{f|n_i}$, with $n_i = n_1, \ldots, n_{\max}$
  c) From $\hat{p}_{n|f}$, find the confidence intervals for n^*, $q_{n|f}^{0.025} \leq n \leq q_{n|f}^{0.975}$ for $f \leq f_b^{\min}$
- **Step 4.** For the selected classifier, use a feature selection algorithm to find $\mathbf{t}^*$ (see Chapter 4). Use $q_{n|f}^{0.025}$ and $q_{n|f}^{0.975}$ to adapt the $n_{\mathrm{m}} \leq n \leq n_{\mathrm{M}}$ search space at each iteration

It is noted that Steps 2.a and 3.a are alike. For the latter, the estimation of $\hat{p}_{f|n_i}$ needs to be accomplished by fitting a Gaussian distribution. The former estimation is unconstrained and for instance, histogram techniques are possible.

3.6 Results with Simulated Data

In order to show the effectiveness of the proposed methods, a set of simulation studies have been carried out. The predictions of the resampling method are compared with the optimal feature subset obtained by an exhaustive search.

Four different models have been considered, all of them using $N = 20$ features (higher number of features makes the exhaustive search unfeasible):

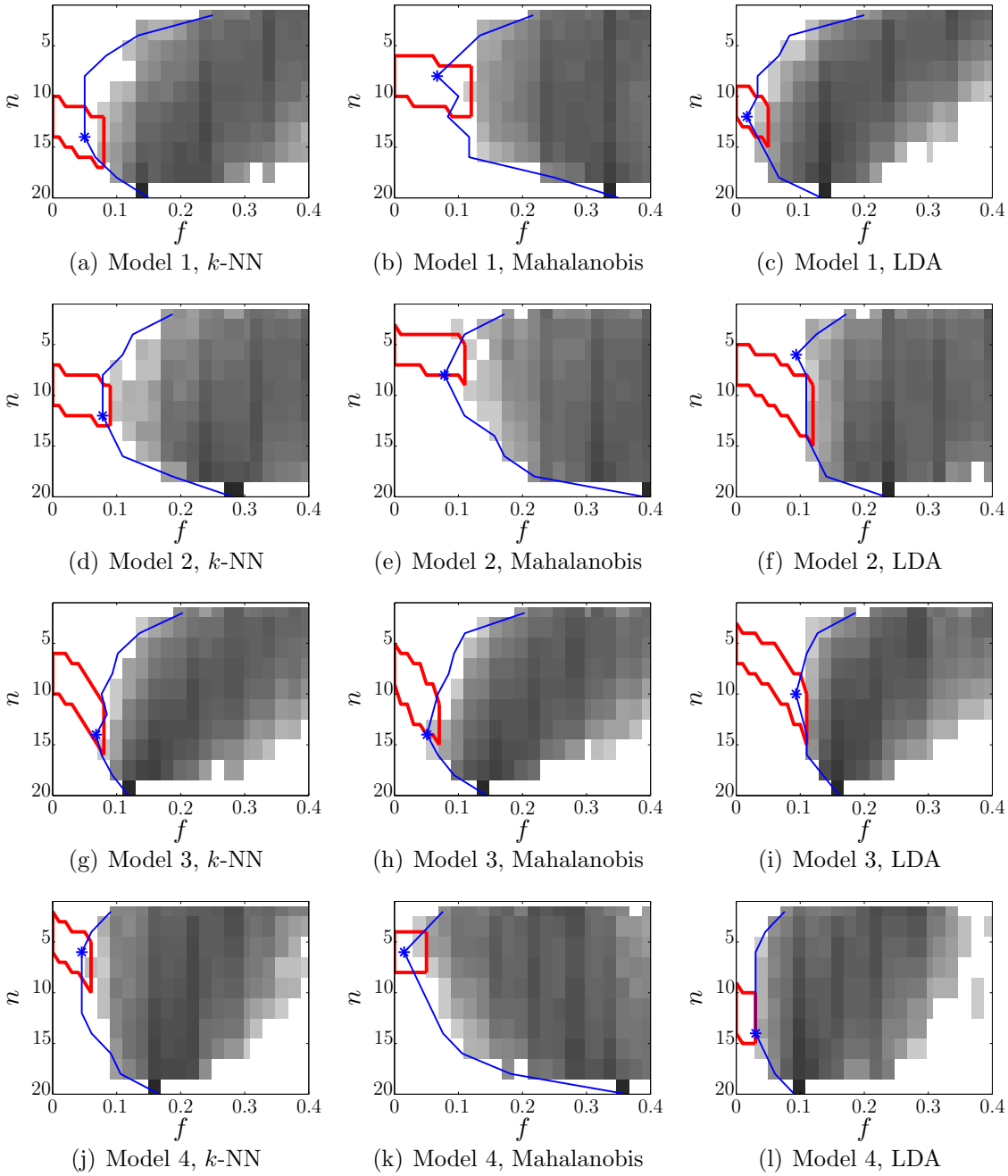
- **Model 1.** Similarly to Model 1 in [59], a 2-class model with Gaussian features whose means are located at $\delta \cdot \mathbf{a}$ and $-\delta \cdot \mathbf{a}$ is employed. The vector $\mathbf{a} = \{a_1, \ldots, a_N\}$ is generated at random from a Gaussian distribution of mean 0 and variance 1. Its magnitude is normalize to $||\mathbf{a}|| = 1$. The features are uncorrelated and their variance is set to 1. The value of δ is chosen such that, for Mahalanobis' classifier, the theoretical misclassification tends to 0.1 as $S \to \infty$.

- **Model 2.** As Model 1, the features are uncorrelated and follow Gaussian distributions of variance 1. The class means are located at $\delta \cdot \mathbf{a}$ and $-\delta \cdot \mathbf{a}$. The mean value a_j follows a uniform distribution between 0.7 and 1 for five out of the 20 features. The mean of the 15 remaining features is generated at random from a Gaussian distribution of mean 0 and variance 1. The value of δ is chosen such that, for Mahalanobis' classifier, the theoretical misclassification tends to 0.2 as $S \to \infty$.

- **Model 3.** As Model 1, the features are Gaussian and the class means are located at $\delta \cdot \mathbf{a}$ and $-\delta \cdot \mathbf{a}$. The vector $\mathbf{a}$ is generated at random from a Gaussian distribution of mean 0 and variance 1. Its magnitude is normalize to $||\mathbf{a}|| = 1$. The correlation of ten out of the 20 features has been set to random values higher than 0.5. The value of δ is chosen such that, for Mahalanobis' classifier, the theoretical misclassification tends to 0.1 as $S \to \infty$.

- **Model 4.** The features follow uncorrelated bimodal Gaussian distributions of different variance values.

Model 1 corresponds to the simplest classification scenario. Model 2 aims for a low optimal number of features in relation to N. Model 3 studies the effect of correlated features in the proposed method. Model 4 studies how the lack of Gaussianity affects $\hat{p}_{f|n_i}$. Other non Gaussian feature distributions (e.g., Weibull and Gamma) have been considered, obtaining similar results to those presented in the sequel.



(a) Model 1, k-NN (b) Model 1, Mahalanobis (c) Model 1, LDA

(d) Model 2, k-NN (e) Model 2, Mahalanobis (f) Model 2, LDA

(g) Model 3, k-NN (h) Model 3, Mahalanobis (i) Model 3, LDA

(j) Model 4, k-NN (k) Model 4, Mahalanobis (l) Model 4, LDA

Figure 3.6: Quality assessment of classifier performance and confidence intervals of the optimal number of features. The curves represent $10 \cdot \log(\hat{p}_{f|n_i})$ for $n_i = 1, \ldots, N$, estimated through histogram techniques from the figure of merit estimates. The estimated 25 % confidence intervals for the optimal number of features restricted to $f \leq f_b^{\min}$ are indicated with a red line. The blue curves show the optimal number of features for each value of n_i. A star is located at the position of the overall optimal number of features n^* and associated f^*. Both the database and the classification approach are indicated for each figure. The scale is common for all figures and spans between -5 dB (white) and 25 dB (black). Although f is defined in the interval $0 \leq f \leq 1$, we focus on the left part of the span.

Twenty data sets have been generated for each model, half of them using $S = 30$ observations and the other half with 60 observations. The number of observations has been chosen deliberately low in order to provoke an optimal dimensionality significantly lower than N. The prior probabilities vary between 0.3–0.7 and 0.5–0.5.

3.6.1 Quality Assessment of Classifier Performance

For each setup, the resampling algorithm has been applied with $N_{\mathrm{B}} = 1000$ samples and three different classifier candidates: k-Nearest Neighbor (k-NN) with $k = 5$, Linear Discriminant Analysis (LDA) and Mahalanobis' classifier. The overall misclassification rate constitutes the figure of merit f, and the 10-fold cross validation has been employed for estimating it from the data.

For one example of each model, the representation of $\hat{p}_{f|n_i}$ in the $f - n$ plane is included in Fig. 3.6. The two first data sets employ $S = 30$ observations and the other two examples correspond to $S = 60$. Each row presents the results of a different simulation model. Choosing one specific $n = n_i$ (i.e., one horizontal line in the image) the distribution of the figure of merit for that specific n_i is obtained. On the other hand, fixing a value for f (i.e., one vertical line in the image) we obtain $\hat{p}_{n|f}$, only lacking the normalization constant A defined in Eq. (3.23). A logarithmic scale has been employed, and the scale spans between -5 dB (white) and 25 dB (black). The distribution of f subject to n_i is approximated by the histogram of $\{f'_{1,n_i}, \ldots, f'_{N_{\mathrm{B}},n_i}\}$, $\hat{p}_{f|n_i} := H_{f|n_i}$. A good classifier will have most of the distribution energy concentrated on the left side.

In general, the variance of $\hat{p}_{f|n_i}$ decreases with n_i. Indeed, for $n_i = N$ a single sample is available and the variance of $\hat{p}_{f|N}$ is 0. In all examples the value of f for $n_i = N$ is higher than $f_b^{\min}$, which confirms that, in general, choosing a subset of features results in a better performance than using all available ones.

The quality assessment has been calculated as indicated in Eq. (3.20) with $\psi = 1$. For the examples in Fig. 3.6, results are included in Table 3.1, so that the three classifier candidates can be compared for each data set. The value of Q for the best classifier has been highlighted.

3.6.2 Optimal Number of Features

The method proposed in Sec. 3.5.2 for predicting the optimal number of features n^* has been applied for the three classifiers. Instead of 95 %, 25 % confidence intervals

Table 3.1: Performance assessment Q, with $\psi = 1$, of a k-NN classifier with $k = 5$, Mahalanobis' classifier and an LDA for the simulated data sets in Fig. 3.6. For each example the best result has been highlighted.

Database	Model 1	Model 2	Model 3	Model 4
k-NN	9.3	7.3	10.72	11.5
Mahal.	6.8	6	**10.73**	8.9
LDA	**12.5**	**7.7**	10.7	**16.5**

Table 3.2: Best figure of merit f^* found by an exhaustive search for the k-NN with $k = 5$, Mahalanobis' and LDA classifiers for the simulated data sets in Fig. 3.6. For each example the best result has been highlighted.

	Model 1	Model 2	Model 3	Model 4
k-NN	0.050	**0.078**	0.068	0.045
Mahal.	0.067	**0.078**	**0.051**	**0.015**
LDA	**0.017**	0.094	0.093	0.03

are employed in order to study the convergence to the real optimal value. The region delimited by $q_{n|f}^{0.375} \leq n \leq q_{n|f}^{0.625}$ and $f \leq f_b^{\min}$ is represented for each data set in Fig. 3.6 by a red line.

The Kullback-Leibler divergence K [60] has been employed to quantify the suitability of the model. It measures the goodness of fit between two distributions by estimating the expected number of extra bits required to code samples from one of the distribution using a code based on the other distribution. Hence, the higher K is, the more different the distributions are. The features of Models 1, 2 and 3 are drawn from Gaussian distributions, so $K(t_j) \approx 0$ bits, $1 \leq j \leq N$. The Kullback-Leibler divergence between the Model 4 features distribution and their fit to a Gaussian is $K(t_j) = 5$ bits in average. By contrast, the Kullback-Leibler divergence between the histogram of $\{f'_{1,n_i}, \ldots, f'_{N_\mathrm{B},n_i}\}$ and their best fit to a Gaussian is around $K(\hat{p}_{f|n_i}) \approx 0.1$ bits (never higher than 0.13 bits), $1 < n_i < N$, for Models 1, 2, and 3 as well as for Model 4. This suggests that there is no strong correlation between the Gaussianity of the features and the Gaussianity of $\hat{p}_{f|n_i}$.

3.6.3 Verification: Optimal Feature Set

Note that a theoretical computation of the confidence intervals is not possible. Even for the simple Model 1, $\hat{p}_{f|n_i}$ diverges from its theoretical error rate due to the small amount of observations that is employed. If a higher amount of observations were employed in order to approach the theoretical value, the curse of dimensionality would not occur and the estimation of the optimal number of features would be unnecessary. Thus, in order to verify the accuracy of the estimated confidence intervals, an exhaustive search has been applied. It finds the overall optimal subset $\mathbf{t}^*$ and its associated f^* and n^*. Results are included in Fig. 3.6. The blue curves indicate the best figure of merit for each n_i. The overall optimum is indicated with a star. The distance between f^* and $f_b^{\min}$ represents the gain of the exhaustive search with respect to selecting the best figure of merit estimate (see Eq. (3.19)). For 209 out of the 240 simulations (three classifier candidates per data set), the optimum falls within the 25 % confidence interval. If 75 % confidence intervals are considered, all simulations show agreement between the predicted confidence intervals and the actual optimal number of features.

The value of f^* can be employed to verify the correctness of the predicted optimal classifier. Table 3.2 shows the value of f^* for all three classifier candidates and for each example in Fig. 3.6. The lowest result for each database is highlighted. For the selection of the best classifier candidate to be correct, the highest Q value in Table 3.1 should coincide with the lowest figure of merit value in Table 3.2. The selection of the best classifier is correct for the first and the third examples. Note, however, that for the second and third data sets, the value of Q is similar for all classifier candidates. In such cases Q does not provide valuable information for selecting the classifier. For $\psi = 1$, the quality assessment of classifier performance is correct 75 % of the times. Most of the mismatches correspond to data sets whose Q values for the different classifier candidates are similar. For $\psi = 0.5$ the values of Q for the different classifier candidates are closer to each other. Moreover, the percentage of mismatches is higher. If $\psi = 2$, the Q values for the different classifier candidates diverge more than for $\psi = 1$, but the percentage of good decisions remains.

Roughly the same rate of error in both the confidence intervals estimation and classifier performance assessment is observed for the four models, also independently of the number of observations and prior probabilities.

Table 3.3: Number of classes C, number of observations S, and dimensionality N, for all databases under consideration. The number of observations has been broken down into the number of observations per class.

Database	PARK	CANC1	CANC2	IONO	MUSK	MULTI
C	2	2	2	2	2	3
S_c	[147 48]	[357 212]	[151 47]	[225 126]	[207 269]	[100 100 100]
N	22	30	32	32	166	76

Table 3.4: Performance assessment Q, with $\psi = 1$, of a k-NN classifier with $k = 5$, Mahalanobis' classifier and an SVM approach with radial basis kernel. For each data set, the best result has been highlighted.

Database	PARK	CANC1	CANC2	IONO	MUSK	MULTI
k-NN	10	24	4	7	6	15
Mahal.	8	22	**4.7**	11	7	14
SVM	**13**	**35**	4.6	**14**	**14**	**18**

3.7 Results with Real Data

The algorithm above has also been applied to six real data sets from the UCI Machine Learning repository [29]: the Parkinsons, Breast Cancer Wisconsin (Diagnostic), Breast Cancer Wisconsin (Prognostic), Ionosphere, Musk (version 1) and Multiple Features data sets. Hereafter, they are referred to as PARK, CANC1, CANC2, IONOS, MUSK and MULTI, respectively. Their number of classes C, number of available features N and number of observations for each class S_c, $1 \leq c \leq C$ are shown in Table 3.3. For the MULTI data set, three out of the ten available classes are considered. Moreover, only 100 out of the available 200 observations per class have been considered.

3.7.1 Quality Assessment of Classifier Performance

Three classifiers have been compared for each data set: the k-NN classifier (see Sec. 3.2.1) with $k = 5$, Mahalanobis' classifier (see Sec. 3.2.2) and an SVM classifier with a radial basis kernel (see Sec. 3.2.4). The overall misclassification rate constitutes the figure of merit f, and the leave-one-out technique has been employed for its estimation from the data.

Table 3.5: Best figure of merit f^* found by the SFS and SFFS feature selection algorithms for the k-NN with $k = 5$, Mahalanobis' and an SVM classifier. For each data set and each feature selection algorithm the best result has been highlighted.

	PARK		CANC1		CANC2	
	SFS	SFFS	SFS	SFFS	SFS	SFFS
k-NN	0.036	0.031	0.021	0.019	0.197	0.157
Mahal.	0.077	0.056	0.021	0.018	**0.157**	0.151
SVM	**0.020**	**0.015**	**0.014**	**0.010**	0.167	**0.136**

Table 3.6: Best figure of merit f^* found by the SFS and SFFS feature selection algorithms for the k-NN with $k = 5$, Mahalanobis' and an SVM classifier. For each data set and each feature selection algorithm the best result has been highlighted.

	IONO		MUSK		MULTI	
	SFS	SFFS	SFS	SFFS	SFS	SFFS
k-NN	0.060	0.059	0.055	0.036	0.007	0.007
Mahal.	0.057	0.046	0.044	0.021	**0.003**	**0.003**
SVM	**0.028**	**0.02**	**0.017**	**0.015**	0.010	0.010

Figs. 3.7 and 3.8 depict $\hat{p}_{f|n_i}$, with $n_i = n_1, \ldots, n_{\max}$, for PARK, CANC1 and CANC2, and for IONOS, MUSK and MULTI, respectively. The distributions are shown in the $f - n$ plane.

The peaking effect or curse of dimensionality is visible for the three examples in Fig. 3.8, but not for every classifier. Indeed, the SVM (third column) seems to be resistant to the curse. Mahalanobis' classifier suffers from it in the three examples (Figs. 3.8(b), 3.8(e) and 3.8(h)), while for the k-NN it is visible only for the IONOS data set (Fig. 3.8(a)).

The quality measure proposed in Eq. (3.20) for $\psi = 1$ is provided in Table 3.4. The value of Q for the best classifier has been highlighted. The curves in Figs. 3.7 and 3.8 match the results in Table 3.4: basically, the lower the f value that is reached, the higher the quality assessment Q that is obtained. Similar Q values indicate that both classifiers reach roughly the same f value, but have no information about the corresponding number of features or the shape of the curve in the $f - n$ plane. For example, for the MUSK database, the k-NN and Mahalanobis' classifiers result in almost the same Q value, 6 and 7 respectively. However, the curves, represented in

Figs. 3.8(d) and 3.8(e) respectively, are significantly different. Both of them reach though, a comparably small f value, which is consistent with their similar Q values.

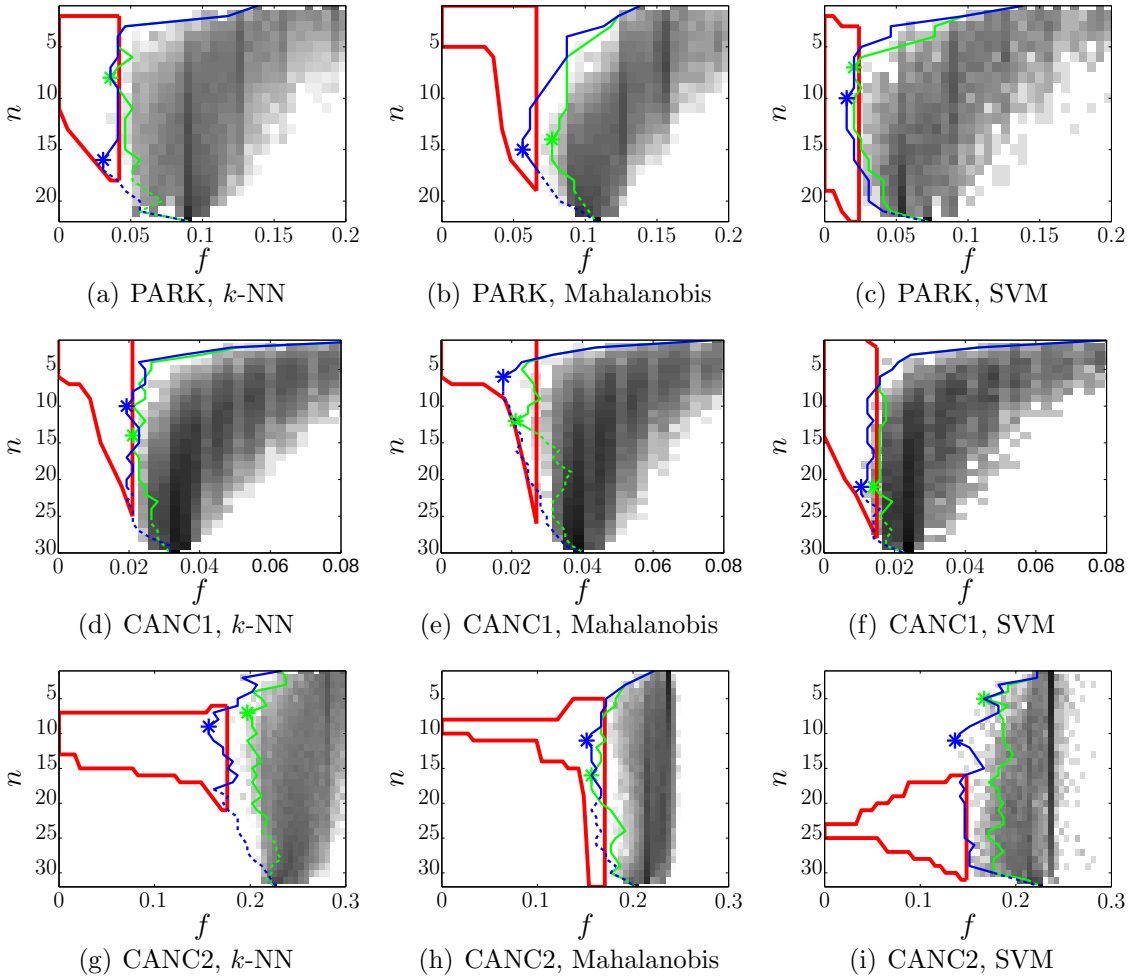


Figure 3.7: Quality assessment of classifier performance and optimal number of features. Each figure shows $10 \cdot \log(\hat{p}_{f|n_i})$ for $n_i = 1, \ldots, N$ for a different database and classification system, estimated through histogram techniques from the figure of merit estimates. The colorbar spans between -5 dB (white) and 25 dB (black). The region where n^* is expected is delimited by a red line. The merit corresponding to the optimal feature sets provided by the SFS and SFFS algorithms are depicted in green and blue, respectively. The pair $\{f^*, n^*\}$ is indicated by a star. For $n > n_{\mathrm{M}}$, dashed lines are employed. Although f is defined in the interval $0 \leq f \leq 1$, we focus on the left part of the span.

3.7.2 Optimal Number of Features

The method proposed in Sec. 3.5.2 for predicting the optimal number of features n^* has been applied to all six examples and all three classifiers. The $f-n$ region delimited

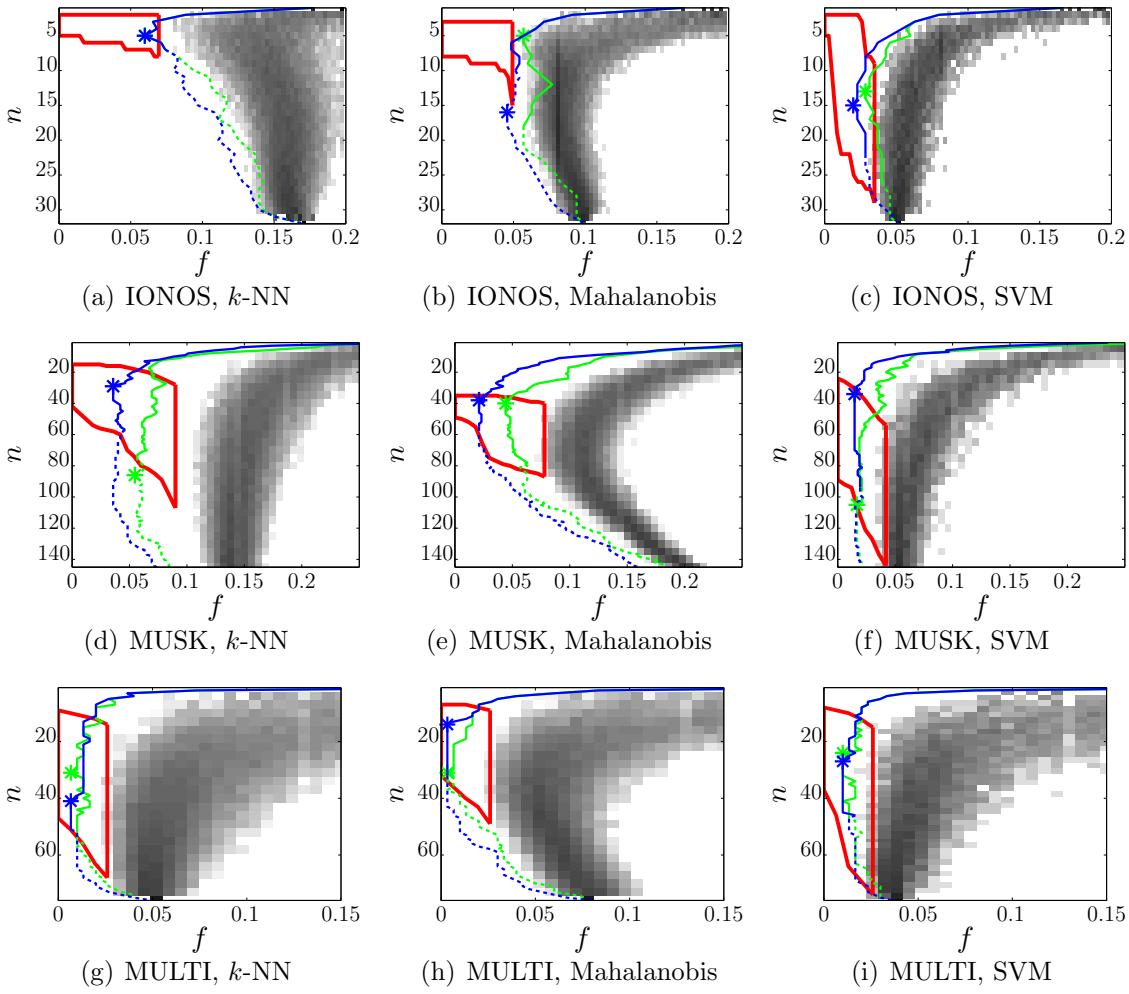


Figure 3.8: Quality assessment of classifier performance and optimal number of features. Each figure shows $10 \cdot \log(\hat{p}_{f|n_i})$ for $n_i = 1, \ldots, N$ for a different database and classification system, estimated through histogram techniques from the figure of merit estimates. The colorbar spans between -5 dB (white) and 25 dB (black). The region where n^* is expected is delimited by a red line. The merit corresponding to the optimal feature sets provided by the SFS and SFFS algorithms are depicted in green and blue, respectively. The pair $\{f^*, n^*\}$ is indicated by a star. For $n > n_{\mathrm{M}}$, dashed lines are employed. Although f is defined in the interval $0 \leq f \leq 1$, we focus on the left part of the span.

by $q_{n|f}^{0.025} \leq n \leq q_{n|f}^{0.975}$ and $f \leq f_b^{\min}$ is indicated by a red line in Figs. 3.7 and 3.8. It represents the confidence region in which the optimal number of features and according figure of merit are to be found with a probability of 95 %. The confidence intervals have been calculated by fitting a Gaussian to the figure of merit estimates, $\{f'_{1,n_i}, \ldots, f'_{N_{\mathrm{B}},n_i}\}$. The narrower the region is in the n direction, the more information it provides. For some data sets, e.g., the IONOS database for the k-NN classifier (Fig. 3.8(a)), $q_{n|f}^{0.975} - q_{n|f}^{0.025} \approx 7 - 2$ is narrow $\forall f \leq f_b^{\min}$ and therefore, the search of the optimal feature subset can be significantly constrained beforehand (see Eqs. (3.24) and (3.25)). On the

other hand, for Mahalanobis' classifier applied to the CANC2 database (Fig. 3.7(h)), $q_{n|f}^{0.975} - q_{n|f}^{0.025}$ is narrow for $f < 0.15$ but otherwise increases up to $N = 32$. Hence, only if for some iteration l of the feature selection algorithm $f_l < 0.15$, the n search space can be reduced to $n_{\mathrm{m}} = 8$ and $n_{\mathrm{M}} = 14$ (for $f_l < 0.1$, n_{M} can be further reduced to 10), as indicated by Eq. (3.26).

3.7.3 Verification: Optimal Feature Set

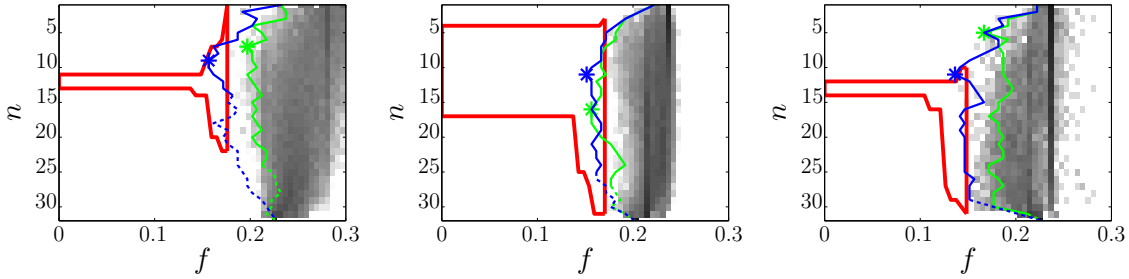
Only an exhaustive search over the feature space guarantees the optimal feature set [18]. However, the number of features and observations of real classification problems are generally too high and, normally, computationally more efficient feature selection algorithms are employed. They provide a suboptimal approach to $\mathbf{t}^*$. In this sequel, two feature selection algorithms, the Sequential Forward Selection (SFS) and the Sequential Floating Forward Selection (SFFS) have been used in order to prove the correctness of both the performance assessment and the estimated $\{f^*, n^*\}$ region. They are described in detail in Chapter 4. For illustration purposes, the algorithms have been applied to all three classifiers (k-NN with $k = 5$, Mahalanobis' and SVM). In a real application though, only the best classifier should be considered. Also for illustration purposes, the n search space has not been limited by the confidence intervals (see Eq. (3.26)) and hence $n_{\mathrm{m}} = 1$ and $n_{\mathrm{M}} = N$.

Both the SFS and SFFS algorithms are sequential and provide a feature set $\mathbf{t}_n^*$ for each n, $1 \leq n \leq N$. The actual n^* is obtained as

$$n^* := \arg\min_n \{f(\mathbf{t}_n^*)\}, \quad 1 \leq n \leq N. \tag{3.27}$$

The value of f corresponding to the optimal feature sets $\mathbf{t}_n^*$ produced by the SFS and the SFFS algorithm for $1 \leq n \leq N$ have been depicted in Figs. 3.7 and 3.8. The stars indicate the value of f^* and the optimal number of features n^*.

The optimal figure of merit found by the feature selection algorithms is higher than $f_b^{\min}$ in four occasions, namely, SFS for PARK with Mahalanobis' (Fig. 3.7(b)), SFS for CANC2 with k-NN and SVM (Figs. 3.7(g) and 3.7(i)), and SFS for IONOS with Mahalanobis' (Fig. 3.8(b)). For these examples, the star is located at the right of the vertical red line, which indicates the position of $f_b^{\min}$. This is not a failure of the proposed method for prediction of the optimal number of features but rather due to the limitations of the SFS method, which is prone to converge to local minima. In such cases, $f_b^{\min}$ should be adopted as f^*. Out of the 32 remaining simulations, the



(a) k-NN (b) Mahalanobis (c) SVM

Figure 3.9: Confidence intervals for CANC2 using a non-parametric approach.

pair $\{f^*, n^*\}$ is located within the region delimited by $q_{n|f}^{0.025} \leq n \leq q_{n|f}^{0.975}$ in 29 cases. It is outside this region for the SFFS result of the SVM classifier applied to CANC2 (Fig. 3.7(i)), the SFFS result of Mahalanobis' classifier applied to IONOS (Fig. 3.8(b)), and the SFS result of the k-NN classifier applied to MUSK (Fig. 3.8(d)). In the last two cases though, the optimal results are very close to the expected region.

The CANC2 $\{f^*, n^*\}$ pair is further away. The values of the quantiles $q_{n|f}^{0.025}$ and $q_{n|f}^{0.975}$ are obviously influenced by the estimated $\hat{p}_{f|n_i}$, which is accomplished by fitting a Gaussian distribution (see Sec. 3.5.2). However, in Fig. 3.7(i) one can see that the histogram of the figure of merit estimates is far from being Gaussian. This mismatch has been quantitatively estimated by the Kullback-Leibler divergence between $\hat{p}_{f|n_i}$ and the histogram of $\{f'_{b,n_i}\}$ for each $n_i = n_1, \ldots, n_{\max}$. The value of K remains below 0.2 bits for all classifiers and data sets except when Mahalanobis' and the SVM classifiers are used on the CANC2 database. In those cases K exceeds 0.6 bits for $n < 7$ and $n < 15$, respectively. As an alternative to the Gaussian fit, let us consider a non-parametric model based on kernel density estimation for the CANC2 data set [61]. A Gaussian kernel has been chosen, and the bandwidth parameter has been set to 0.05. The new confidence intervals are illustrated in Fig. 3.9. The SVM confidence intervals seem significantly more suitable than those in Fig. 3.7(i). The difference between the Gaussian fit and the non-parametric confidence intervals for the k-NN and Mahalanobis' classifiers is noticeable for very small values of f but negligible in the region of interest.

Note that, in a real application, the number of iterations could be reduced to $n_{\mathrm{M}} = q_{n|f^*}^{0.975}$ (see Eq. (3.26)). In order to show the computational cost that can be saved, dashed curves are employed for $n > n_{\mathrm{M}}$. For instance in Fig. 3.7(e), n_{M} is initialized to $n_{\mathrm{M}} = 29$ (see Eq. (3.25)), but after it can be lowered to $n_{\mathrm{M}} = 13$ and $n_{\mathrm{M}} = 16$ for the SFFS and the SFS algorithms, respectively. On the other hand, the value of f^* for the CANC2 data set in combination with Mahalanobis' classifier (Fig. 3.7(h)) does not

reach $f \leq 0.15$ for SFS algorithm and, therefore, n_{M} cannot be constrained. In most of the examples though, n_{M} can be significantly reduced during the feature selection process.

The value of the optimal f^* provided by the SFS and SFFS algorithms allows for the verification of the quality assessment obtained in Sec. 3.7.1. In order to be consistent, the smallest f^* should correspond to the best classification system. Tables 3.5 and 3.6 summarize the value of f^* for all databases and classifiers. The smallest value for each database, for both the SFS and the SFFS feature selection algorithms, has been highlighted. The comparison with Table 3.4 is straightforward: the highest Q should correspond to the smallest f^*. There is an agreement for all databases and for both feature selection algorithms, with the exception of CANC2 for the SFFS result and MULTI for both the SFS and SFFS. Note that the value of Q for Mahalanobis' and the SVM classifiers are almost identical. This is not the case for the MULTI data set.

Chapter 4

Feature Selection

Due to the curse of dimensionality (see Sec. 3.1), given a classification problem where a finite number of observations S is available and a feature vector $\mathbf{t}$ of dimension N is provided for each observation s, $1 \leq s \leq S$, it is advantageous to select a subset of n^* features, $\mathbf{t}^* = \{t_1^*, t_2^*, \ldots, t_{n^*}^*\}$, with $n^* < N$. While the estimation of n^* has been tackled in the previous chapter, this chapter considers the estimation of the actual elements in $\mathbf{t}^*$. Only an exhaustive search over the feature space guarantees the optimal feature subset [18]. However, the size of the feature candidate database N, makes this task prohibitive in most cases and therefore, search algorithms providing suboptimal estimates for $\mathbf{t}^*$ are necessary.

Many feature selection methods exist in the literature (see [20, 28] for a review). Although some controversy has arisen in the last years [62, 63], it is generally accepted that the Sequential Forward Floating Selection (SFFS) algorithm [64] provides the best performance. The SFFS method is based on the simpler Sequential Forward Selection (SFS) method [65]. While the latter sequentially adds features to the optimal subset, the former allows as well for removal. The algorithms suffer from limitations that undermine their performance, e.g., the nesting of feature subsets (the best n-feature subset does not necessarily contain the best subset of $n-1$ elements). In this thesis, an extension of both algorithms is proposed. It alleviates their limitations by storing the best D options (with $D > 1$) instead of choosing a single feature at each step. Thus, the extended algorithms are called D-SFFS and D-SFS, respectively. They outperform the standard SFFS and SFS methods, respectively, at the expenses of a higher computational cost, which scales linearly with D. Nevertheless, their computational cost is in general significantly reduced when the search space is constrained to the confidence intervals for optimal dimensionality proposed in Sec. 3.5.2.

For a given feature selection method, the provided $\mathbf{t}^*$ depends on the available data and the selected evaluation criterion, that is, the figure of merit f. According to the latter, the feature selection algorithms can be divided into three main groups: filter, wrapper and hybrid models. The filter model does not employ any specific classifier to evaluate the performance of the different feature subsets. It is estimated, for example, by measuring the mutual information between features and between features and classes [66]. The wrapper model, by contrast, calculates the performance of a given feature subset by utilizing a certain classifier, i.e., it computes the figure of merit

f (e.g. probability of misclassification). The hybrid model combines both previous approaches. While filter models are generally faster, they also tend to provide poorer results. In this thesis, only the wrapper approach is considered.

The chapter is organized as follows. Sec. 4.1 describes the SFS and SFFS methods. Their extended D-SFS and D-SFFS versions are presented in Secs. 4.2 and 4.3, respectively. Their performance has been evaluated on six standard databases of the UCI Machine Learning repository [29] (already used in Chapter 3). Results are presented in Sec. 4.4.

4.1 Standard Methods: SFS & SFFS

The SFS algorithm [65] is initialized with the best single feature, that is, with the feature that optimizes the figure of merit f,

$$t_1^* = \arg\min_{t_j \in \mathbf{t}} f(t_j), 1 \leq j \leq N. \tag{4.1}$$

Subsequently, the SFS algorithm adds one feature at a time that in combination with the already selected ones optimizes f:

$$\mathbf{t}_n^* := \{\mathbf{t}_{n-1}^*, t_n^*\}, \tag{4.2}$$

with $t_n^* = \arg\min_{t_j \in \mathbf{t} \setminus \mathbf{t}_{n-1}^*} f(\{\mathbf{t}_{n-1}^*, t_j\})$. The algorithm stops when n reaches a preselected value n_{M}, with $n_{\mathrm{M}} \leq N$. The output of the SFS algorithm is not $\mathbf{t}^*$, but a set $\mathbf{t}_n^*$, $1 \leq n \leq n_{\mathrm{M}}$. The optimal subset $\mathbf{t}^*$ is obtained as

$$\mathbf{t}^* := \arg\min_{\mathbf{t}_n^*} \{f(\mathbf{t}_n^*)\}, \quad 1 \leq n \leq n_{\mathrm{M}}. \tag{4.3}$$

Its corresponding figure of merit, $f(\mathbf{t}^*)$, is denoted by f^*.

The SFS algorithm requires a high amount of computation per iteration and therefore, it is more appropriate to constraint the search space to $n < n_{\mathrm{M}}$. If the resampling method presented in Chapter 3 has been applied in order to estimate confidence intervals for the optimal number of features, the value of n_{M} can be constrained as suggested by Eq. (3.26). In this case, n_{m} is not required. Note that the iteration index l in Eq. (3.26) is now denoted by n, since it coincides with the dimension of $\mathbf{t}_n^*$.

Analogously to the SFS algorithm, the Sequential Backward Selection (SBS) [67] is initialized with the whole feature set $\mathbf{t}$ and removes one feature at a time. The SBS

search space can be limited to $n \geq n_{\mathrm{m}}$. The value of n_{m} is updated after each iteration as indicated by Eq. (3.26). By contrast with the SFS algorithm, the SBS method does not require n_{M}.

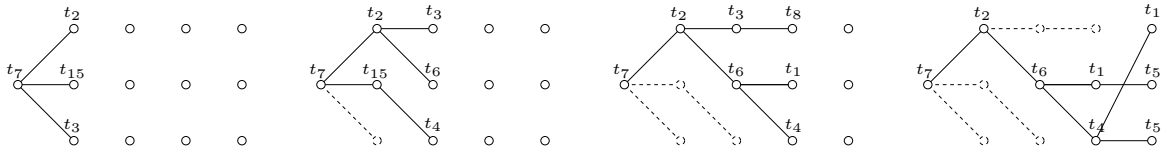
The main drawback of both the SFS and SBS algorithms lies in the so-called nesting problem, that is, the optimal 3-subset is not necessarily contained in the optimal 4-subset, and so on [28]. The SFFS algorithm [64] tries to overcome this issue. After initializing the feature set, $\mathbf{t}_0^* := \emptyset$, the SFFS repeats the following steps until the size of $\mathbf{t}_n^*$ reaches $n = n_{\mathrm{M}}$:

- **Step 1** (SFS). Find $t_n^* = \arg\min_{t_j \in \mathbf{t} \setminus \mathbf{t}_{n-1}^*} f(\{\mathbf{t}_{n-1}^*, t_j\})$ and update $\mathbf{t}_n^* := \{\mathbf{t}_{n-1}^*, t_n^*\}$, $n := n + 1$
- **Step 2** (Conditional SBS).
  - Find $t_n^{**} = \arg\min_{t_j \in \mathbf{t}_n^*} f(\mathbf{t}_n^* \setminus t_j)$
  - If $f(\mathbf{t}_n^* \setminus t_n^{**}) \leq f(\mathbf{t}_{n-1}^*)$ then
    update $\mathbf{t}_{n-1}^* := \mathbf{t}_n^* \setminus t_n^{**}$
    $n := n - 1$ and go to Step 2
    else go to Step 1

The value of n_{M} is updated after each iteration as indicated by Eq. (3.26). The output of the algorithm is again a set $\mathbf{t}_n^*$, $1 \leq n \leq n_{\mathrm{M}}$. The optimal subset $\mathbf{t}^*$ is estimated as indicated by Eq. (4.3). Note that the index l, which appears in Eq. (3.26) to denote the iteration number, has not been employed in the formulation of the SFFS above. Instead, n denotes the current number of elements in $\mathbf{t}_n^*$, disrespectfully of the number of iterations required to obtain it.

4.2 D-SFS

In this thesis, a novel modification of the SFS algorithm alleviating the nesting issue is proposed. Instead of choosing the best possible feature at a time, the best D ($D > 1$) candidates are kept in a matrix $\mathbf{T}(d, n)$ whose structure can be represented as a tree of D branches (see Fig. 4.1 and Table 4.1). While n, $1 \leq n \leq n_{\mathrm{M}}$, refers to the feature index in $\mathbf{T}$, d, $1 \leq d \leq D$, indicates the branch. Note that the optimal 5-subset of the example, $\{t_7, t_2, t_6, t_4, t_1\}$, does not contain the best 4-subset, $\{t_7, t_2, t_3, t_8\}$.



(a) $n = 2$ features (b) $n = 3$ features (c) $n = 4$ features (d) $n = 5$ features

Figure 4.1: Illustration of the D-SFS working principle, with $D = 3$. Notice that the same feature can be added more than once at the same iteration n as long as the corresponding branches do not coincide already at previous iterations. The evolution of the matrix $\mathbf{T}(d, n)$ is included in Table 4.1.

Table 4.1: Evolution of the matrix $\mathbf{T}(d, n)$ for the example in Fig. 4.1. At each stage, the first row corresponds to the lowest merit, i. e., the optimal subset for that dimensionality.

		2 Features		**3 Features**			**4 Features**				**5 Features**				
Branch	1	t_7	t_2	t_7	t_2	t_3	t_7	t_2	t_3	t_8	t_7	t_2	t_6	t_4	t_1
	2	t_7	t_{15}	t_7	t_2	t_6	t_7	t_2	t_6	t_1	t_7	t_2	t_6	t_1	t_5
	3	t_7	t_3	t_7	t_{15}	t_4	t_7	t_2	t_6	t_4	t_7	t_2	t_6	t_4	t_5

The algorithm is formulated as follows. The matrix $\mathbf{T}$ is initialized for all branches with the best single feature: $\mathbf{T}(d, 1) := t_1^* \ \forall d$, where $t_1^* = \arg\min_{t_j \in \mathbf{t}} f(t_j)$, $1 \leq j \leq N$. Then, n is increased iteratively until it reaches n_{M}:

- **Step 1.** Compute the figure of merit that stems from adding each possible t_j feature to each branch d: $f_{n,d,j} = f(\{\mathbf{T}(d, \mathbf{n}), t_j\})$, where $\mathbf{n} = \{1, \cdots, n\}$ $\forall t_j \in \mathbf{t} \setminus \mathbf{T}(d, \mathbf{n})$, $\forall d$
- **Step 2.** Sort $f_{n,d,j}$ for all d and all j, in an ascending fashion and store the result in $\hat{f}_{n,d,j}$. Create the vectors $\hat{\mathbf{d}}$ and $\hat{\mathbf{t}}_j$ with the values of d and t_j corresponding to $\hat{f}_{n,d,j}$
- **Step 3.** Update the tree:
  - Select the branches that correspond to the best merit $\mathbf{T}(d, \mathbf{n}) := \mathbf{T}(\hat{\mathbf{d}}(d), \mathbf{n})$, $1 \leq d \leq D$
  - Add the best feature to each branch: $\mathbf{T}(d, n+1) := \hat{\mathbf{t}}_j(d)$, $1 \leq d \leq D$
  - $n := n + 1$

Different branches correspond to different values of f and therefore to different quantiles $q_{n|f}^{0.975}$. The value of n_{M} is updated after each iteration (see Eq. (3.26)) according to

Table 4.2: Example of evolution of the matrix $\mathbf{T}(d, n)$ for the D-SFFS algorithm with $D = 3$.

		3-SFS it. n				**SBS it. n**				**3-SFS it. n + 1**			
Branch	1	t_7	t_2	t_3	t_8	t_7	t_2	t_3	t_8	t_7	t_2	t_3	t_8
	2	t_7	t_2	t_6	t_1	t_7	t_2			t_7	t_2	t_8	
	3	t_7	t_2	t_6	t_4	t_2	t_6	t_4		t_2	t_6	t_4	

		SBS it. n + 1				**3-SFS it. n + 2**				**SBS it. n + 2**			
Branch	1	t_7	t_2	t_3	t_8	t_7	t_2	t_3	t_8	t_7	t_2	t_3	t_8
	2	t_2	t_8			t_2	t_8	t_1		t_2	t_8	t_1	
	3	t_2	t_6	t_4		t_2	t_6	t_4		t_2	t_6	t_4	

		3-SFS it. n + 3				**SBS it. n + 3**				**3-SFS it. n + 4**				
Branch	1	t_7	t_2	t_3	t_8	t_7	t_2	t_3	t_8	t_2	t_8	t_1	t_6	t_5
	2	t_2	t_8	t_1	t_7	t_2	t_8	t_1	t_7	t_7	t_2	t_3	t_8	t_1
	3	t_2	t_8	t_1	t_6	t_2	t_8	t_1	t_6	t_2	t_8	t_1	t_6	t_4

the f value of the best alternative, that is, the first row of $\mathbf{T}$, $f(\mathbf{T}(1, \mathbf{n}))$. Once the iterative process is finished, the best alternative is selected

$$\mathbf{t}_n^* = \mathbf{T}(1, \mathbf{n}), \quad 1 \leq n \leq n_{\mathrm{M}}. \tag{4.4}$$

The optimal feature subset and corresponding f^* are then found according to Eq. (4.3).

The computational complexity of the D-SFS algorithm increases linearly with D.

4.3 D-SFFS

The performance of the SFFS algorithm can also be improved by tracking several branches, i.e., alternatives. Due to the SBS step in the SFFS algorithm, some branches might consist of less features than others at a given moment. Hence a different n_d is employed for each branch d. A vector $\mathbf{a}$ that lists the so-called active branches is defined. A branch d is active if it has the least amount of features, i.e., if $n_d = \min\{n_1, ..., n_D\}$. The state of the branches is updated only after the SBS step.

We initialize $\mathbf{T}$ as for D-SFS: $\mathbf{T}(d,1) := t_1^* \; \forall d$, where $t_1^* = \arg\min_{t_j \in \mathbf{t}} f(t_j)$, $1 \leq j \leq N$. At the beginning all branches are active: $\mathbf{a} = \{1, ..., D\}$ and $n_d = 1 \; \forall d$. Then, we iterate until $\min\{n_1, ..., n_D\} = n_{\mathrm{M}}$:

- **Step 1.** D-SFS considering only the active branches
- **Step 2.** Conditional SBS (Step 2 of SFFS) for each active branch
- **Step 3.** Update $\mathbf{a}$: $d \in \mathbf{a}$ iff $n_d \leq \min\{n_1, ..., n_D\}$

Note that the tree representation used for the D-SFS is no longer possible since, for instance, the first feature might be deleted for one of the branches but not for the others. The matrix representation is preferred. An example is included in Table 4.2. Its corresponding iterations are detailed as follows:

- **it. n:**
  - **3-SFS:** $\mathbf{a} = \{1, 2, 3\}$.
  - **SBS:** branches 2 and 3 loose features, $\mathbf{a} = \{2\}$.
- **it. n + 1:**
  - **3-SFS:** only on branch 2.
  - **SBS:** only branch 2 is active, and it looses one feature, $\mathbf{a} = \{2\}$.
- **it. n + 2:**
  - **3-SFS:** only on branch 2.
  - **SBS:** only branch 2 is active, and it does not loose any feature, $\mathbf{a} = \{2, 3\}$.
- **it. n + 3:**
  - **3-SFS:** on branches 2 and 3.
  - **SBS:** all branches are active but none looses features, $\mathbf{a} = \{1, 2, 3\}$.
- **it. n + 4:**
  - **3-SFS:** on all branches.
  - ...

As for the D-SFS, n_{M} is updated according to the best option (first branch) at each

iteration. Also, the best alternative is selected after the iterative process as indicated by Eq. (4.4).

The computational complexity of the D-SFFS algorithm increases linearly with D.

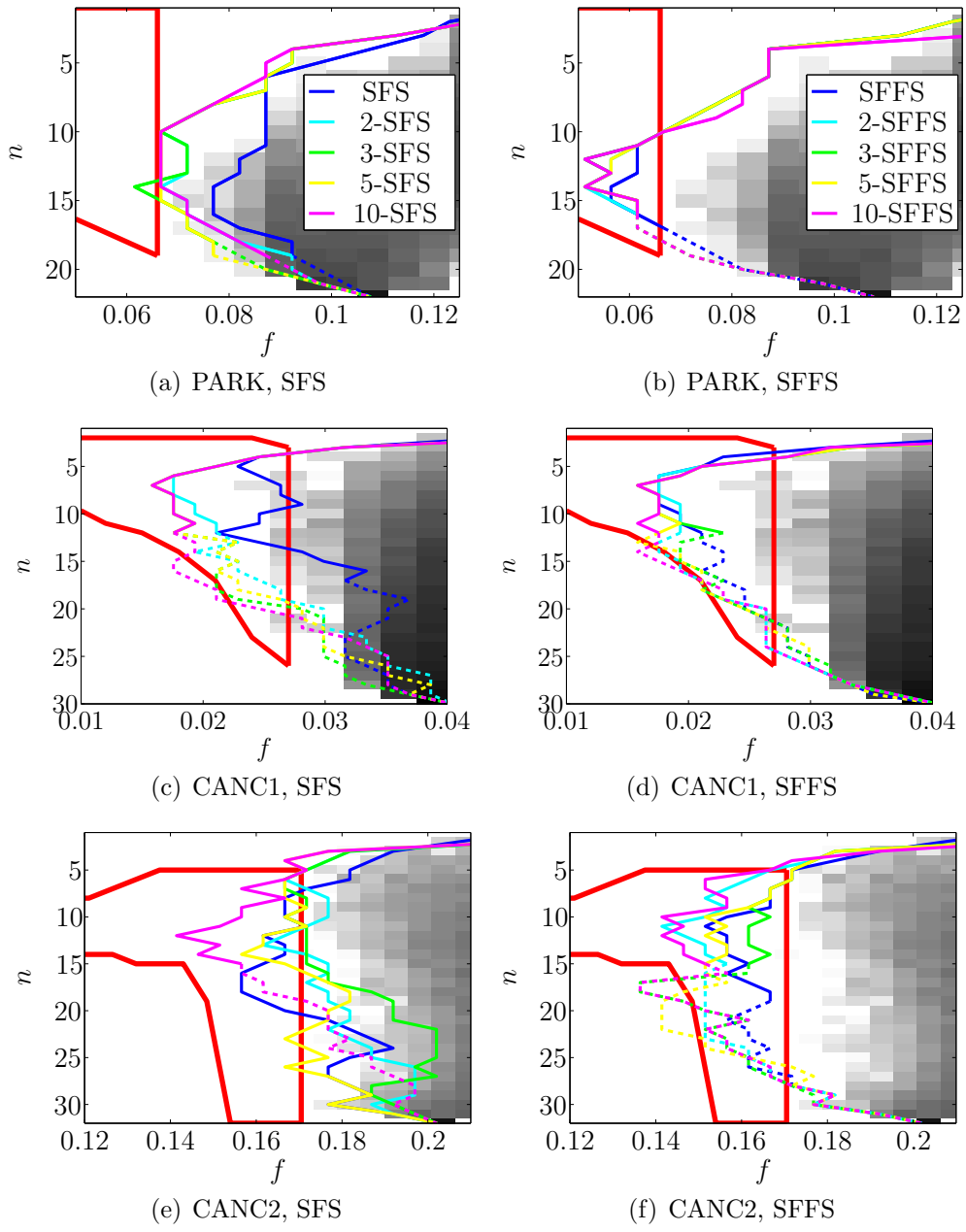
4.4 Performance Evaluation

Both the D-SFS and the D-SFFS algorithms have been tested on the six databases from the UCI Machine Learning repository [29] already used in Chapter 3 (see Table 3.3 for details). Mahalanobis' classifier is utilized and the parameter D takes the values $\{2, 3, 5, 10\}$. Again, the misclassification rate is chosen as figure of merit. Results are depicted in Figs. 4.2 and 4.3. For illustration purposes, the region where the pair $\{f^*, n^*\}$ is expected (see Chapter 3), is delimited by a red line. Note that most of the curves reach their minimum within this region. In order to show the computational cost that can be saved by limiting the search space to the confidence intervals of n^*, dashed curves are employed for $n > n_{\mathrm{M}}$. For all data sets, the D-SFS and the D-SFFS methods outperform the SFS and SFFS algorithms for all or most D values. There are some cases, however, where the extended algorithm provides a worse performance than the standard method. See, for example, the 2-SFS and 3-SFS results for the CANC2 data set in Fig. 4.2(e). Hence, an increase of D does not guarantee a better result, although it makes it more probable.

The average performance is summarized in Fig. 4.4, which illustrates the relative improvement of the optimal figure of merit, Δf^*, that the D-SFS and D-SFFS algorithms provide with respect to the SFS f^* as a function of D. The curves have been calculated by averaging the results from the six UCI databases. For $D \geq 3$, the D-SFS f^* is almost 20 % lower than the SFS f^*. Analogously, the D-SFFS reduces the value of f^* about 10 % with respect to the SFFS algorithm. Increasing the value of D beyond three provides, on average, no significant improvement for either the D-SFS or the D-SFFS.

Note that the optimal SFFS figure of merit outperforms the SFS f^* by more than 15 %. For $D \geq 3$, however, the D-SFS outperforms the SFFS algorithm. Furthermore, the 3-SFS algorithm is computationally more efficient than the SFFS: while the SFFS is typically considered to be five and ten times slower than the SFS, the 3-SFS is only three times as costly.

The computational cost of the SFFS algorithm is in general between six and ten times higher than the SFS cost. The execution time of both the D-SFS and the D-SFFS

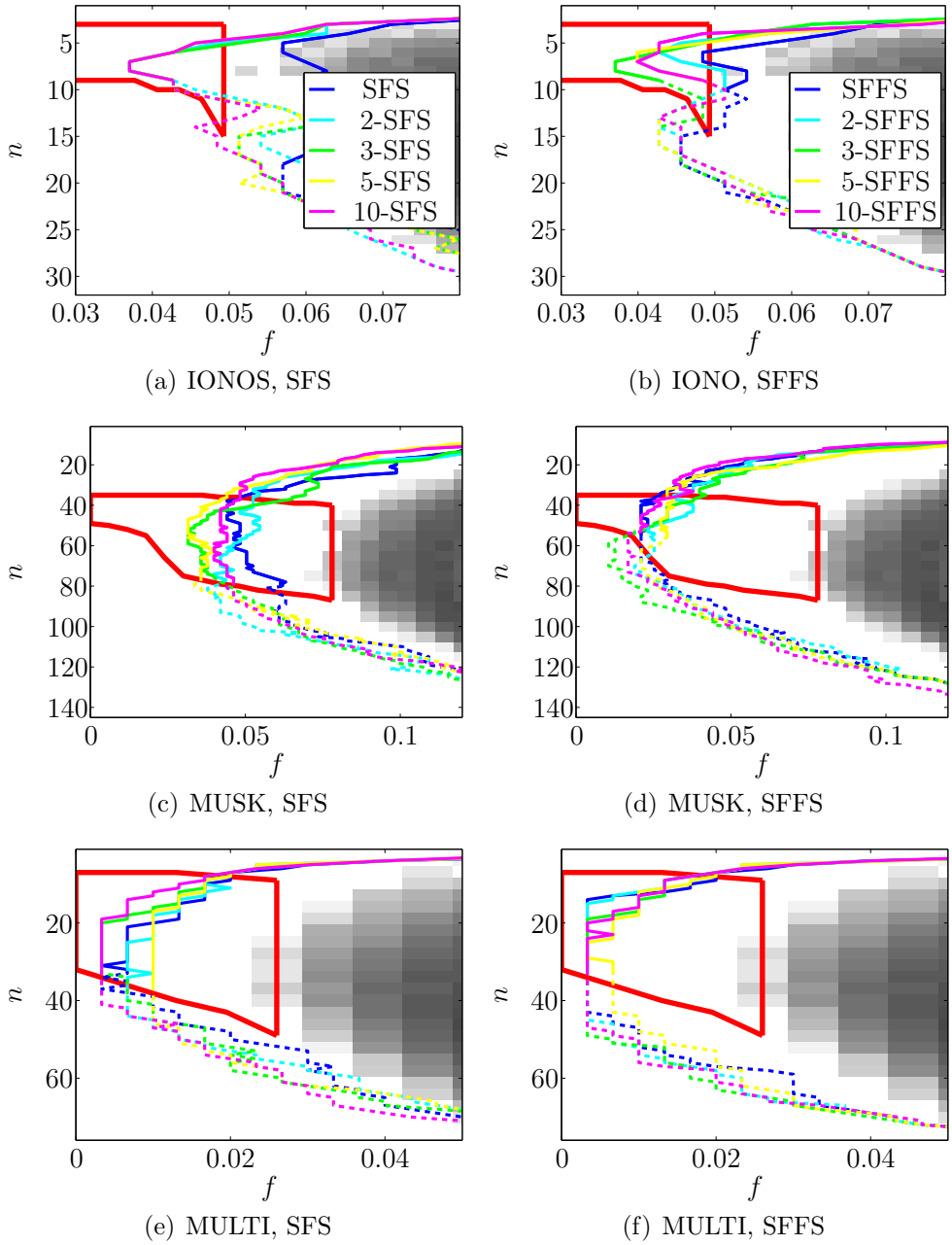


(a) PARK, SFS (b) PARK, SFFS

(c) CANC1, SFS (d) CANC1, SFFS

(e) CANC2, SFS (f) CANC2, SFFS

Figure 4.2: Performance of the D-SFS and D-SFFS algorithms. The figures on the left column show the performance of the D-SFS algoritm, $D = \{2, 3, 5, 10\}$, for the PARK, CANC1 and CANC2 databases. For comparison, the SFS performance has been included. On the right column, the D-SFFS performance is depicted. For illustration purposes, the distribution of f subject to the number of features, $10 \cdot \log(\hat{p}_{f|n_i})$, is included in the background. The red line indicates the region where n^* is expected according to the algorithm presented in Chapter 3. For $n > n_{\mathrm{M}}$, dashed lines are employed.



(a) IONOS, SFS (b) IONO, SFFS

(c) MUSK, SFS (d) MUSK, SFFS

(e) MULTI, SFS (f) MULTI, SFFS

Figure 4.3: Performance of the D-SFS and D-SFFS algorithms. The figures on the left column show the performance of the D-SFS algoritm, $D = \{2, 3, 5, 10\}$, for the IONOS, MUSK and MULTI databases. For comparison, the SFS performance has been included. On the right column, the D-SFFS performance is depicted. For illustration purposes, the distribution of f subject to the number of features, $10 \cdot \log(\hat{p}_{f|n_i})$, is included in the background. The red line indicates the region where n^* is expected according to the algorithm presented in Chapter 3. For $n > n_{\mathrm{M}}$, dashed lines are employed.

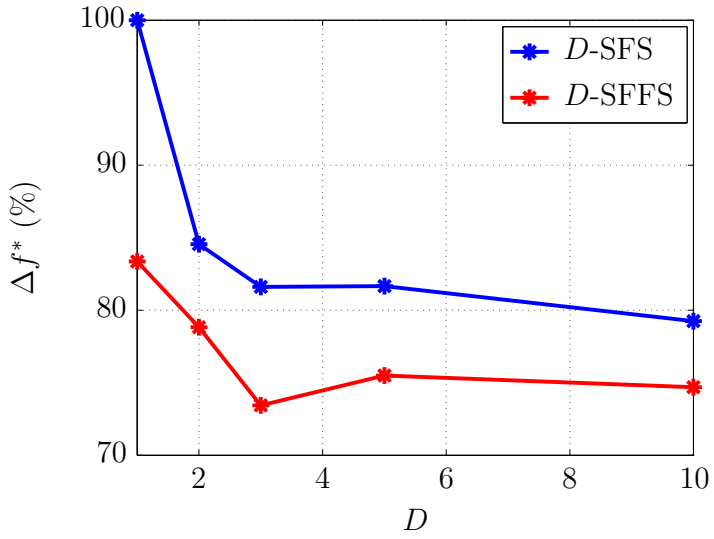


Figure 4.4: Relative decrease of f^* achieved by the D-SFS and D-SFFS algorithms with respect to the standard SFS (represented at $D = 1$). The curves have been calculated by averaging the performance of the six UCI data sets.

scales linearly with D. In Sec. 9.5, the computational time of the algorithms for the mine hunting ADAC system design is examined.

Chapter 5

Conclusions and Future Work

The first part of this thesis has tackled some fundamental problems of pattern recognition: the selection of the optimal classifier, the estimation of the optimal number of features and the selection of the optimal feature subset. The two first issues have been addressed by a resampling algorithm. Furthermore, an extension of two well-known feature selection algorithms has been proposed.

A summary and the main conclusions of the work performed in the first part of this thesis are provided in Sec. 5.1. Finally, Sec. 5.2 provides an outlook for possible future work.

5.1 Conclusions

5.1.1 Quality Assessment of Classifier Performance and Optimal Number of Features

Before addressing the resampling method proposed in this thesis, which fulfills the twofold purpose of quality assessment for classifier performance and estimation of the optimal number of features, some fundamental concepts have been defined. First, the curse of dimensionality is described. It is responsible for the performance degradation from a certain size of the feature subset and therefore, it accounts for the necessity of estimating the optimal features subset. Subsequently, the theory behind the four classification systems employed in this thesis is presented: the k-Nearest Neighbor (k-NN), Mahalanobis' classifier, the Linear Discriminant Analysis (LDA) and the Support Vector Machines (SVM). The LDA approach is employed only in the second part of the thesis. Finally, the resampling fundamentals are described. This method allows for statistical inference of parameters of random variables when few realizations are available or too little is known about their statistics.

Based on resampling techniques, a novel algorithm for estimating the probability distribution of the figure of merit of a classifier (typically the misclassification rate) subject to the dimensionality of the feature set has been proposed. It serves as a framework

for the design of classification systems, addressing two fundamental issues: the choice of the classifier and the size of the optimal feature subset.

In order to quantify the performance quality of a classifier, the distribution of its figure of merit is multiplied by a weighting function and then integrated, in such a way that the result is higher for classification systems whose figure of merit distribution concentrates in the small values. Unlike traditional methods, this algorithm allows for the performance evaluation of classifiers independently of any specific feature subset.

The estimation of the optimal number of features by traditional methods typically assumes Gaussianity for the features and a common covariance matrix for the different classes. The resampling method requires none of these assumptions. It estimates the empirical distribution of the number of features conditional on the figure of merit, and it allows for computing its confidence intervals. Since the distribution needs to be extrapolated to values of the figure of merit where no estimate is available, histogram techniques, which may be employed for the classifier quality assessment application of the method, are no longer valid. Thus, a parametric model extrapolating the distribution for small values of the figure of merit is required. In this thesis, a Gaussian model is adopted. Note that while traditional approaches assume Gaussianity for the features, the resampling method adopts a Gaussian model for the distribution of the figure of merit.

The effectiveness of the algorithm for its twofold purpose has been tested on 80 synthetic databases and on six sets of real data of the UCI Machine Learning repository. An exhaustive search of the overall optimal feature subset has been performed in order to verify the correctness of the resampling method predictions. For 75 % of the synthetic data sets the selection of the optimal classifier is correct. The optimal number of features is located within the estimated 25 % confidence intervals in 87 % of the cases and within the 75 % confidence intervals for all examples. The method shows the same average performance for Gaussian and uncorrelated features as for non Gaussian and correlated features.

Regarding the real data, the performances of three classifier candidates, k-NN, Mahalanobis' and an SVM classifiers, have been compared according to the proposed quality assessment. In order to verify the performance of the proposed techniques, two feature selection algorithms have been applied to the data for the three classifier candidates. The performance and size of the actual optimal feature set have been compared with the estimates of the resampling algorithm. Regarding the classifier performance assessment, only one significant mismatch has occurred. Among the 36 results, again only one significant mismatch between the predicted and the actual dimensionality exists.

The lack of Gaussianity of the figure of merit is likely to be the reason, and a correct estimation of the intervals is obtained by means of a non-parametric approach.

5.1.2 Feature Selection

After choosing the optimal classification system, the selection of the optimal feature subset leads to a further performance improvement. Two iterative feature selection methods have been regarded in this thesis, the Sequential Forward Search (SFS) and the Sequential Forward Floating Search (SFFS). The latter is typically considered as the best feature selection method. The computational cost of the former is significantly smaller. The SFS algorithm iteratively adds features to the best feature subset, optimizing the performance with respect to the figure of merit. The SFFS algorithm allows for removal of features as well. This alleviates the main limitation of the SFS, namely, the nesting effect.

Since only an exhaustive search in the feature space guarantees finding the overall optimal subset, both the SFS and the SFFS provide suboptimal approximations, leaving place for improvement. In this thesis, an extension of the methods is proposed. It yields better results than the standard algorithms at the expenses of a higher computational cost. The extended algorithms, referred to as D-SFS and the D-SFFS, store D candidate optimal subsets. This fights the nesting problem of the SFS algorithm, and improves the performance of the SFFS as well. The computational cost of the algorithms increases linearly with D.

The methods have been tested on the same six data sets employed for the resampling method verification with $D = \{2, 3, 5, 10\}$. Mahalanobis' classifier has been used. For $D > 3$, the D-SFS and the D-SFFS algorithms improve the figure of merit about 20 % and 10 % in average with respect to the standard SFS and SFFS, respectively. The D-SFFS algorithm is in average better than the D-SFS algorithm for any value of D. However, the D-SFS algorithm is better than the SFFS algorithm for $D > 3$, and its computational cost is smaller (generally, it is considered that the SFFS algorithm is between five and ten times slower than the SFS). Therefore, it is more advantageous to employ the D-SFS with $D > 3$ than the SFFS algorithm, both from the performance and the computational cost points of view.

The main limitation of the D-SFS and D-SFFS methods is the fact that, although increasing D augments the probability of finding a better feature subset, it does not guarantee it.

As a second verification of the resampling method for the prediction of the optimal feature set dimensionality, the confidence intervals that the resampling algorithm provide have been compared with the optimal dimensionality that the D-SFS and D-SFFS algorithms find. A remarkable agreement has been observed.

5.2 Future Work

5.2.1 Quality Assessment of Classifier Performance and Optimal Number of Features

The selection of the best classifier candidate based on the quality assessment for classifier performance fails in 25 % of the simulated examples and leaves place for improvement. Most of the mismatches between the selected optimal classifier and the classifier providing the actual optimal feature subset occur for data sets whose performance assessment is similar for several classifier candidates. A more sophisticated quality measure overcoming this limitation is desired.

When the resampling method is employed for estimation of the optimal number of features, there exists a need to extrapolate the value of the figure of merit distribution beyond the values provided by the figure of merit estimates. Fitting a Gaussian distribution has proved as an efficient approach, since the figure of merit distribution is close to Gaussian in most cases. However, when the distribution is not Gaussian, an alternative is required. A non-parametric approach based on kernel density estimation has been proposed. The estimation of a suitable bandwidth parameter remains as future work. Other parametric or non-parametric models might be investigated as well.

5.2.2 Feature Selection

The D-SFS and the D-SFFS outperform in average the standard SFS and SFFS, respectively. Also in average, the performance of the algorithms saturates for $D > 3$. However, it might happen that, for a specific case, a high D value provides a worse results than a lower one. Moreover, it is also possible that, for a specific case, employing $D > 3$ provides a significant performance improvement. A criterion that, with a reasonable computational cost, helps the pattern recognition practitioner to choose an advantageous value for D given a particular database, would increase the usefulness of the algorithms.

Part II

ADAC for Mine Hunting using SAS Imagery

Chapter 6

Introduction

Like the kids looking for shells along the beach, in the application of mine hunting, the identification and classification of the objects present on the seabed have traditionally been performed by human operators. The first attempts of Computer Aided Detection and Classification (CAD/CAC) employed sidescan sonar images [68]. However, it is the high resolution provided by Synthetic Aperture Sonar (SAS) technology (see [69] and references therein) that has increased the interest in ADAC for mine hunting applications. Fig. 6.1 shows a SAS image snapshot with several man made objects.

Sidescan sonar systems use an array of hydrophones to scan the seabed in a narrow beam fashion [70]. The reconstructed images present a resolution that is not only limited by the array length but also decreases with range. SAS technology overcomes these limitations: the sonar system moves along a straight line and pulses are sent at different positions. The reflected signals are recorded and further combined in order to reconstruct the scene. Hence, a synthetic array is built up, allowing for an increased resolution. Furthermore, it can be demonstrated that the resolution is range independent [71]. Synthetic aperture techniques were first used for radar applications [72]. Its adaptation to the sonar environment is not straightforward, mainly due to the need for very precise navigation error compensation, which is significantly more challenging for sonar than for radar.

Regarding ADAC systems for mine hunting, two main approaches are typically adopted, either template fitting or feature description. Template fitting applies a set of allowed transformations to a given template, so that it matches the region of interest. The distance between the deformed template and the object is measured in some norm and the object is classified accordingly. In this thesis, the feature description approach has been adopted. Besides the design issues tackled in the first part of this thesis, which are common to all ADAC systems independently of the application, there exist a few considerations specific to mine hunting. They are described in the following.

The detection of objects on the reconstructed SAS image is performed by a segmentation algorithm. Its result determines to a great extent the performance of the complete system, since any lost information cannot be retrieved. Sonar images are typically divided into three regions: the object highlights, their shadows and the background. Plenty of clutter is segmented together with the objects of interest. For instance, the

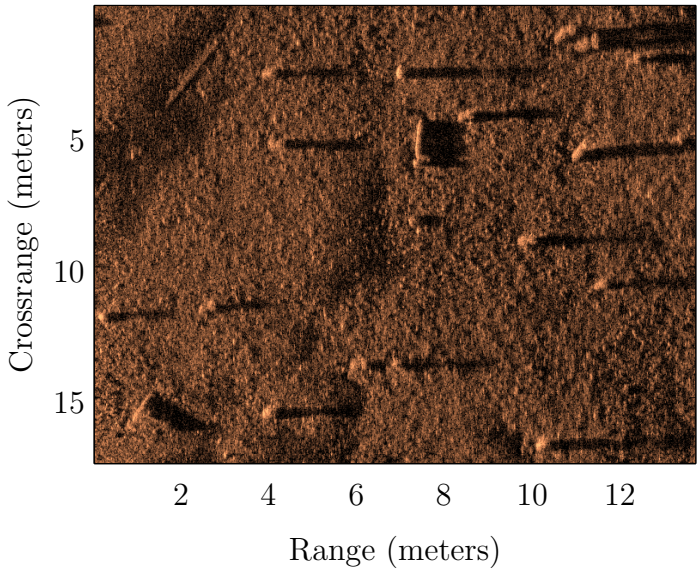


Figure 6.1: Snapshot of a SAS image showing several cylindrical and spherical man made objects. The sonar system moves along the crossrange direction, following a path to the left of the scanned seabed. Some areas of the seabed remain hidden behind the objects and are not illuminated by the ultrasound energy. They originate shadows, i.e., the dark areas of the image to the right of each object. The object highlights correspond to the lighter areas of the image. Note that the shadows of the objects are, indeed, more prominent than the object highlights. Both range and crossrange are measured in meters. The resolution of a pixel is 2.5 cm $\times$ 2.5 cm for the images used in this thesis.

dark seabed area in the center of Fig. 6.1 will, most likely, be segmented as shadow. Therefore, not only mine classes are to be considered, e.g., spherical or cylindrical mine, but more importantly, a clutter class is required. In fact, the objective of an ADAC system for mine hunting is the detection and correct classification of all mines, while keeping a fairly low false alarm rate (clutter classified as mine). The distinction between different kinds of mines, although desirable and addressed in this thesis, is secondary.

SAS images exhibit a fairly high resolution (2.5 cm $\times$ 2.5 cm for the SAS images employed in this thesis), but the very nature of the seabed irremediably hinders the segmentation task. For example, the presence of sand ripples or rocks constitutes especially challenging scenarios [73]. Furthermore, the speckle noise, caused by the coherent processing of backscattered signals from multiple distributed targets, is inherent to both radar and sonar technologies [71, 74]. Finally, the orientation of the object of interest with respect to the sonar antenna might be such that the intensity of the returned echo is too weak for the object highlight to be accurately reconstructed. This effect is far more remarkable in sidescan than in SAS imagery [75]. Therefore, traditional sidescan ADAC systems rely on the shadows of the objects rather than on their highlights.

Let us now consider the feature design. Good features exhibit significantly distinct values for mine and clutter objects, increasing the so-called class separability. They are computed from either the shape of the segmented object –shadow and highlight– or from the statistical properties of the sonar image. A good feature should, moreover, be invariant to possible changes of position of the objects. Consider as features the length of the highlight projected along the crossrange direction and along the direction of its major axis. For cylindrical objects, the former reaches its minimum when the cylinder is parallel to the range direction and its maximum when the object is parallel to the crossrange direction. On the contrary, the length of the highlight measured along the major axis of the cylinder is invariant to the object position. Therefore, the latter is a more appropriate feature than the former.

6.1 State of the Art

Due to its strategic relevance for military applications, the published works in the area of mine hunting often present some limitations. Firstly, although SAS mine hunting is an active research field, few studies are available, and very few show results on real data. Furthermore, the publications are often opaque, making difficult the reproduction of the algorithms. Finally, instead of presenting unified ADAC systems, most works focus on a specific part of the system, e.g., segmentation or feature extraction. However, the interaction between the different parts of the system must be considered in order to optimize the system performance.

The noisy character of the sonar images needs to be regarded by the segmentation algorithms. For this reason, simple methods such as thresholding fail to provide high quality results. An effective segmentation algorithm [76] for sonar images utilizes a Markov Random Fields model of the image and the Iterative Conditional Modes (ICM) algorithm. If its initialization is accurate, it provides reasonable segmentation results for reasonably challenging scenarios. The ICM algorithm has been combined with the Active Contours (AC) [77] to improve the performance.

Template fitting for sidescan sonar images has been considered in [15, 78–80], and in [81–83] for SAS applications. The performance of these methods depends to a great extent on the sophistication of the templates. Best results are obtained when the templates are generated by a simulator of 3D models of the mines [84].

To the knowledge of the author, there exists no published feature selection ADAC system for mine hunting using SAS images. By contrast, sidescan sonar ADAC systems

based on feature schemes have been employed in several studies with successful results [14, 85–89]. So far, the focus has been on descriptors of the shadow shape [90–92]. In [13, 79], Fourier descriptors are considered and in [91] normalized central moments are used. An alternative approach is to focus on the statistical properties of an image. In [93], the mean and variance of the different regions are considered. The kurtosis and skewness are used for detection and classification purposes in [94, 95]. Also the difference of SNR between the different regions has been regarded [68].

A common approach to maximize the performance of mine hunting ADAC systems is the fusion of several simpler systems [14, 88, 96–98]. It was introduced by [96] with the purpose of reducing the false alarm rate. Both [14] and [88] combine three algorithms, and nine are employed in [97]. Generally, each of the simple systems works with one type of features, e.g., Fourier coefficients of the shadow shape. The corresponding classification system is chosen heuristically, namely, no systematic comparison of classification systems is accomplished. Furthermore, all extracted features are considered, that is, no feature selection algorithm is employed to optimize the performance. Alternatively, the fusion of multiple views of the same object can be exploited [99, 100].

6.2 Contributions

In the following the main contributions of this thesis to the field of mine hunting are listed:

- **Segmentation:** Two main contributions are provided. On the one hand, an initialization scheme for the ICM algorithm is proposed, which improves the final segmentation results with respect to standard approaches. On the other hand, the min-cut/max-flow algorithm [101] is applied for the first time for segmentation of sonar images. Its performance is compared with the well-established ICM and AC segmentation algorithms.
- **Feature Extraction:** Instead of employing a single type of features (as most existing works do), e.g., Fourier coefficients, a collection of feature types is extracted: statistical features, geometrical features for both the shadow and the highlight of the objects, normal central moments, principal components, Fourier coefficients, etc. Several of these features, namely, some geometrical shadow and highlight features and the statistical features, are novel to this thesis. They are designed to remain invariant to changes in the position of the object and, moreover, can deal with poor segmentation scenarios.

- **Classification:** Instead of combining different non-optimal simple classification systems (algorithm fusion), a single optimized algorithm is proposed. Thus, the novel resampling method proposed in Chapter 3 is applied to compare a set of classifiers. By doing so, it is assured that the selected system is the most appropriate to the feature space at hand, but avoiding constraints to a specific feature subset. Besides, the resampling algorithm provides the optimal number of features, which reduces the search space of the feature selection algorithms.
- **Feature Selection:** Unlike most existing mine hunting methods, which include all extracted features in the feature set, the extended versions of the SFS and SFFS algorithms are applied in order to estimate the optimal feature subset. This allows for a significant improvement of the system performance.

6.3 Overview of Part II

The second part of the thesis is devoted to the design of an ADAC system for mine hunting based on SAS technology. The proposed algorithms are tested on two extensive databases of images. The first database, called SAS1 in the following, consists of over 57,000 m^2 of SAS images with more than 400 man made objects (spheres and cylinders). The images have been generated with a Vision600 system from ATLAS UK. Visual inspection and classification by several experts was used to generate the ground truth. The second database, denoted by SAS2, comprises around 180 snapshots of mines (cylinders, truncated cones and wedge-shaped objects). The ground truth is known *a priori*, since the objects were placed on the seabed for this purpose. This database has been collected using a synthetic aperture sonar mounted on the MUSCLE autonomous underwater vehicle by the NATO Undersea Research Center (NURC), in the framework of a NATO project for mine countermeasures [102][1].

Chapter 7 presents three segmentation algorithms for sonar images: ICM, AC and min-cut/max-flow. Segmentation results are compared at the end of the chapter but indeed, a more meaningful comparison is provided by their classification performance, which is considered in Chapter 9.

Each object detected in the SAS images is characterized by a set of features, which is described in Chapter 8. A combination of statistical and geometrical features, both for the shadow and the highlight of the objects is considered.

[1]This work is part of a collaboration with ATLAS ELEKTRONIK GmbH. The experiments presented in this thesis were performed at the company site in Bremen, Germany.

Once the data set of segmented objects is constructed and the corresponding feature sets are calculated, the design framework proposed in Part I is employed in order to achieve an optimal design of an ADAC system for mine hunting. First, the resampling method estimates the optimal classifier and the optimal number of features. After, the D-SFS and D-SFFS algorithms provide the actual elements of the optimal feature subset for the optimal classifier. This design approach has been applied to both available SAS databases. The obtained results are presented in Chapter 9. Since the performance of the first database leaves place for improvement, an alternative configuration, based on a cascade of binary classifiers, is investigated. Finally, a comparison of all three segmentation algorithms according to their classification performance is included.

The conclusions and outlook for future work are summarized in Chapter 10.

Chapter 7

SAS Image Segmentation

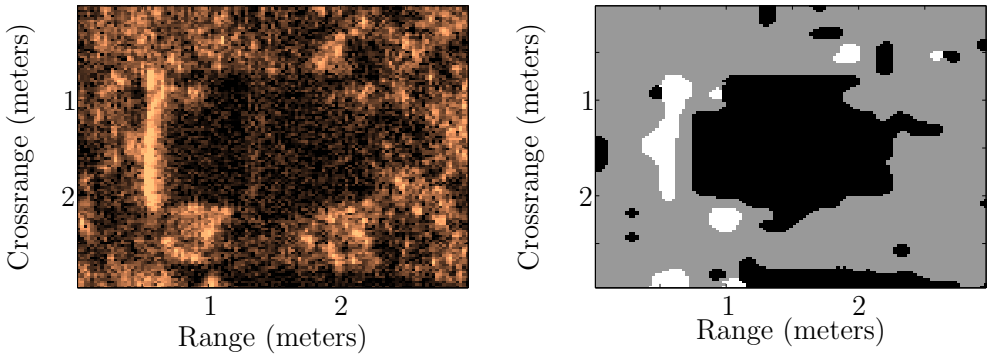
Segmentation is the process of partitioning an image into several regions [103]. A label is assigned to each pixel, so that all pixels with the same label belong to the same region. While equally labeled pixels are similar with respect to some characteristic, e.g., color or intensity, pixels with different labels significantly differ with respect to the same characteristic.

The objective of segmenting an image is its simplification, so that it is easier to analyze. A typical application of image segmentation is the location of objects. This is as well the application considered in this thesis. Subsequently, a set of features is extracted for each object, and they are classified accordingly. Hence, if the segmentation is poor, the features characterizing the object shape will also be poor and therefore, a wrong classification is more likely to happen.

Figs. 7.1(a) and 7.1(b) show an SAS image snapshot and its segmentation, respectively. A cylindrical object and its shadow are observed. Three different regions are considered in sonar image segmentation: the highlights of the objects in the scene, `hlt`, their shadows, `sdw`, and the seabed or background, `bkg`. The objects might be mines, but also physical features of the terrain such as rocks or sand ripples, which constitute clutter. In this thesis, any element that produces a shadow is considered to be an object.

In the literature, there exist several approaches to unsupervised segmentation of sonar images. Some of them are based on simple clustering techniques such as histogram thresholding [91] or fuzzy K-means [104, 105]. These models perform well for flat seabeds with high SNR, but fail for more complex environments such as sand ripples. Markov Random Fields (MRF) [106] have proved to be a valuable model able to cope with such conditions. It considers not only the intensity of the pixels in the image, but also their neighborhood relations.

MRF were employed as a model for sonar images in [107], in order to reduce the noise of the segmented image previously obtained by a simple clustering algorithm. It has also been used in the context of seabed reconstruction from raw sonar data [108,109]. MRF were introduced as an image model for sidescan sonar image segmentation in [110,111] and later used in [76, 77], where they are combined with the Iterative Conditional



(a) SAS sonar image of a cylindrical object. The pixels of lower intensity correspond to the shadow of the object while the object highlight intensity is higher than the background.

(b) Segmentation result. The label for the background pixels `bkg` is depicted in grey, the shadow label `sdw` in black and the highlight label `hlt` in white.

Figure 7.1: SAS sonar image and segmentation result.

Estimation (ICE) algorithm [112] and the Iterative Conditional Modes (ICM) [106]. The ICM algorithm performs, in fact, the segmentation. It requires a set of parameters, which are previously estimated by the ICE algorithm.

Another algorithm that has demonstrated its value in the segmentation of sonar image is the Active Contours (AC) or Statistical Snakes [113]. It has been used in combination with *a priori* information on the relative position of the highlight and shadow regions, the so-called Cooperative Statistical Snakes [77]. The contour of the shadow region is constrained by the position of the corresponding highlight, and the other way around. While this approach provides high quality segmentation results of man made objects, it is prone to produce a high false alarm rate, since randomly shaped clutter regions tend to be segmented similarly to man made objects.

Other segmentation methods have been used for different purposes. For instance, the level set method [114,115] has shown its value for detection of the seabed texture, e.g., sand ripples.

All referred algorithms were proposed for sidescan sonar applications. In this thesis, the ICM algorithm and the AC are applied to SAS imagery. In order to avoid high false alarm rates, no *a priori* position information is considered by the AC. Besides, a min-cut/max-flow algorithm that, as the ICM algorithm, is based on a MRF representation of the image, is proposed in this thesis. The three algorithms require an initialization of the segmentation, which determines to a great extent the final result. A comparison of initialization methods for the ICM algorithm is accomplished in Sec. 7.2.2. As

reference, the implementation proposed in [77] has been used. The initialization of both the min-cut/max-flow and the AC algorithms is based on the ICM segmentation result.

This chapter is organized as follows. In Sec. 7.1, the MRF model, which serves as a basis for both the ICM and the min-cut/max-flow segmentation algorithms, is described. The former is introduced in Sec. 7.2 and a description of the latter is provided in Sec. 7.3. While the ICM is the most extended segmentation algorithm for sonar images, the min-cut/max-flow algorithm is applied for the first time to sonar imagery in the framework of this thesis. The AC algorithm has also been considered (see Sec. 7.4). Sec. 7.5 shows segmentation results of the SAS1 database for all proposed algorithms, and Sec. 7.6 studies their computational cost. Due to confidentiality reasons, no image of the SAS2 data set is displayed in this thesis.

7.1 Markov Random Fields

Let us express a sonar image $\mathbf{Y}$ associated with a lattice $\mathcal{L}$, as a vector $\mathbf{y} = \{y_i, i \in \mathcal{L}\}$, where y_i denotes the intensity of pixel i. The label field $\mathbf{x} = \{x_i, i \in \mathcal{L}\}$ is the 'ground truth' that needs to be recovered, where each pixel has one of the possible labels, $\{\texttt{sdw}, \texttt{hlt}, \texttt{bkg}\}$, assigned. The MRF model consists of the two fields $(\mathbf{y}, \mathbf{x})$. According to Bayes' theorem, the posterior probability density function (pdf) of the label field $\mathbf{x}$ given the sonar image $\mathbf{y}$ corresponds to the expression

$$p_{\mathbf{x}|\mathbf{y}} = p_{\mathbf{x}} \cdot p_{\mathbf{y}|\mathbf{x}}, \tag{7.1}$$

where $p_{\mathbf{x}}$ is the Markovian *a priori* probability and $p_{\mathbf{y}|\mathbf{x}}$ is the likelihood function of the image. Both are described in this section.

Given a MRF model of an image, the optimal estimate for $\mathbf{x}$ is the one that corresponds to the global maximum of Eq. (7.1). Its calculation is, in most of the cases, computationally prohibitive. Both the ICM algorithm (see Sec. 7.2) and the min-cut/max-flow algorithm (see Sec. 7.3) approximate the global maximum by a local one, $\hat{\mathbf{x}}$.

7.1.1 Markovian Probability

Given a pixel i, its neighborhood $\mathcal{M}_i$ is formed by a set of pixels such that $i \notin \mathcal{M}_i$, and $\forall i' \in \mathcal{M}_i$, $i \in \mathcal{M}_{i'}$. Each pair $\{i, i'\} \mid i' \in \mathcal{M}_i$ is known as clique. Several neighborhood

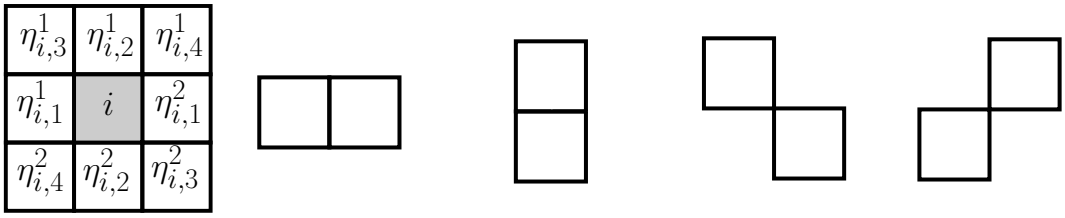


Figure 7.2: A pixel i and its second order neighbors; associated cliques of type 1, 2, 3 and 4.

systems are commonly used in image modeling [103]. For this application the second order neighborhood system is chosen. Fig. 7.2 shows such a neighborhood configuration and its associated cliques. The first clique type relates the pixel i with the neighbors to its right and left ($\eta^1_{i,1}$ and $\eta^2_{i,1}$), the second one models its dependency with the neighbors above and below ($\eta^1_{i,2}$ and $\eta^2_{i,2}$), the third and forth one relates it with the pixels in the first ($\eta^1_{i,3}$ and $\eta^2_{i,3}$) and second ($\eta^1_{i,4}$ and $\eta^2_{i,4}$) diagonals, respectively. Hence, the neighborhood of pixel i is defined by $\mathcal{M}_i = \{\eta^1_{i,1}, \eta^2_{i,1}, \eta^1_{i,2}, \eta^2_{i,2}, \eta^1_{i,3}, \eta^2_{i,3}, \eta^1_{i,4}, \eta^2_{i,4}\}$.

According to the Hammersley-Clifford theorem [116, 117], there is a one-to-one equivalence between MRF and the so-called Gibbs Random Fields, which have an associated Gibbs distribution. This kind of representation is very convenient for modeling the *a priori* probability p_{x_i}, that is, the dependency of a pixel label x_i with the labels of its neighbors $\mathcal{M}_i$. A random field $\mathbf{x}$ has a Gibbs distribution with respect to a neighborhood system $\mathcal{M}$ if and only if its joint distribution can be expressed as

$$p_{x_i} = \frac{1}{Z} e^{-G(x_i)}, \tag{7.2}$$

where Z is a normalizing constant and $G(x_i)$ is an energy function that corresponds to the expression

$$G(x_i) = \mathbf{\Theta}_i^{\mathrm{T}} \mathbf{\Omega}_x = (\Theta_{i,1}, \Theta_{i,2}, \Theta_{i,3}, \Theta_{i,4}) \cdot \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{pmatrix}. \tag{7.3}$$

Each β_j describes the neighborhood relations of a pixel with its neighbors of clique type j, and

$$\Theta_{i,j} = 2 - \delta[x_i - x(\eta^1_{i,j})] - \delta[x_i - x(\eta^2_{i,j})], \quad j = 1, \cdots, 4, \tag{7.4}$$

where $\delta(\cdot)$ is Kronecker's delta and $x(\eta^1_{i,j})$ and $x(\eta^2_{i,j})$ refer to the labels of the neighbor pixels $\eta^1_{i,j}$ and $\eta^2_{i,j}$, respectively. While $\mathbf{\Theta}_i$ can be computed directly from the estimated label field $\hat{\mathbf{x}}$, the parameter vector $\mathbf{\Omega}_x$ needs to be estimated. The method proposed in [118] has been employed.

Including *a priori* knowledge about the topology of the sonar images into the algorithm improves its performance. First of all, highlight regions are always located to the left of shadow regions. Secondly, the typical size of expected highlight regions is known for a given range. Both effects can be taken into account in the Gibbs energy of Eq. (7.3) [119].

7.1.2 Likelihood Function

Assuming that y_i, $\forall i \in \mathcal{L}$, are conditionally independent, the likelihood function for the image $\mathbf{y}$ is given by

$$\begin{aligned} p_{\mathbf{y}|\mathbf{x}} &= \prod_{i \in \mathcal{L}} p_{y_i|x_i} \\ &= \prod_{\{y_i|x_i=\mathtt{bkg}\}} p_{\mathtt{bkg}}(y_i) \cdot \prod_{\{y_i|x_i=\mathtt{sdw}\}} p_{\mathtt{sdw}}(y_i) \cdot \prod_{\{y_i|x_i=\mathtt{hlt}\}} p_{\mathtt{hlt}}(y_i), \end{aligned} \tag{7.5}$$

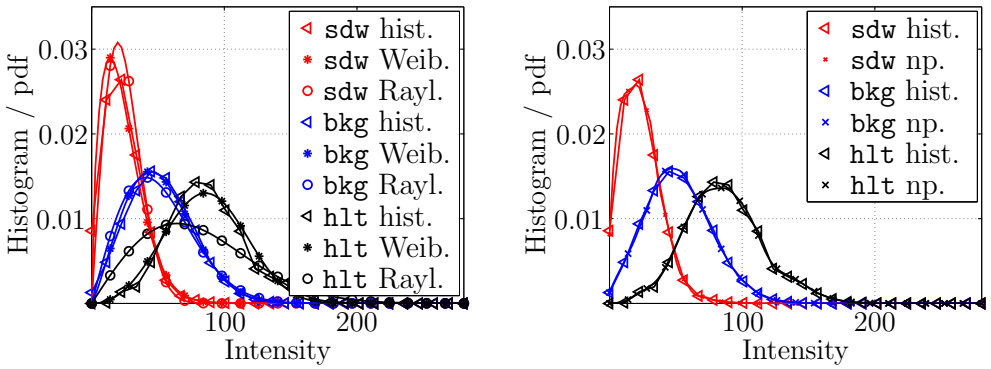
where $p_{\mathtt{bkg}}$, $p_{\mathtt{sdw}}$ and $p_{\mathtt{hlt}}$ are the pdfs of the corresponding regions. Typically, they are unknown and need to be estimated from the available data. Parametric models are traditionally employed. Furthermore, a non-parametric approach based on Kernel Density Estimation (KDE) is proposed in this thesis. In the following, the most commonly accepted parametric models for sonar imagery and the proposed non-parametric approach are presented.

7.1.2.1 Parametric Approach

Two parametric models are commonly used to model the distribution of the pixel intensity y_i in sonar images: the Rayleigh [120] and the Weibull distributions [119, 121], $\mathcal{R}(\alpha)$ and $\mathcal{W}(\xi, \xi')$, respectively. While α refers to the single parameter than defines the Rayleigh distribution, ξ and ξ' are the scale and shape parameters of the Weibull distribution, respectively. Indeed, the Rayleigh distribution equals the Weibull distribution for $\xi' = 2$. Both distributions are defined only for $y_i \geq 0$, which fits the characteristics of sonar images, whose intensity is typically quantified between 0 and 255.

The model parameters are estimated by maximization of the likelihood function [122]. If the Weibull distribution is chosen

$$(\xi'_{\mathtt{sdw}}, \xi_{\mathtt{sdw}}) = \arg\max_{\xi',\xi} \prod_{\{y_i|x_i=\mathtt{sdw}\}} \hat{p}_{\mathtt{sdw}}(y_i) = \arg\max_{\xi',\xi} \prod_{\{y_i|x_i=\mathtt{sdw}\}} \frac{\xi'}{\xi}\Big(\frac{y_i}{\xi}\Big)^{\xi'-1} \cdot \exp\Big(-\frac{y_i}{\xi}\Big)^{\xi'}, \tag{7.6}$$



(a) Histograms, estimated Weibull and Rayleigh pdfs for the different regions of the SAS image in Fig. 7.1(a) as segmented in Fig. 7.1(b).

(b) Histograms and non-parametric pdfs for the different regions of the SAS image in Fig. 7.1(a) as segmented in Fig. 7.1(b).

Figure 7.3: Likelihood function estimation.

where $\hat{p}_{\texttt{sdw}}$ is the estimation of $p_{\texttt{sdw}}$. Analogously, $\xi'_{\texttt{bkg}}$, $\xi_{\texttt{bkg}}$, $\xi'_{\texttt{hlt}}$ and $\xi_{\texttt{hlt}}$ are estimated for the `bkg` and `hlt` regions, which yields the parameter vector $\mathbf{\Omega}_y = (\xi_{\texttt{sdw}}, \xi'_{\texttt{sdw}}, \xi_{\texttt{bkg}}, \xi'_{\texttt{bkg}}, \xi_{\texttt{hlt}}, \xi'_{\texttt{hlt}})$.

If the Rayleigh distribution is chosen,

$$\alpha_{\texttt{sdw}} = \prod_{\{y_i|x_i=\texttt{sdw}\}} \hat{p}_{\texttt{sdw}}(y_i) = \arg\max_{\alpha} \prod_{\{y_i|x_i=\texttt{sdw}\}} \frac{y_i}{\alpha^2} \cdot \exp\left(\frac{-y_i^2}{2\alpha^2}\right) \tag{7.7}$$

and analogously for `bkg` and `hlt`, which yields the parameter vector $\mathbf{\Omega}_y = (\alpha_{\texttt{sdw}}, \alpha_{\texttt{bkg}}, \alpha_{\texttt{hlt}})$.

Fig. 7.3(a) shows the estimated pdfs of the SAS image in Fig. 7.1(a), as segmented in Fig. 7.1(b). For comparison, the histograms are depicted.

7.1.2.2 Non-Parametric Approach

Although both the Rayleigh and, specially, the Weibull distribution suits the available SAS images, a non-parametric model is robust to possible changes of the image statistics. Furthermore, the additional computational cost is negligible.

Given the set of independent and identically distributed samples $\{y_i|x_i = \texttt{sdw}\}$, the kernel density approximation of its pdf is

$$\hat{p}_{\texttt{sdw}}(y) = \frac{1}{N_{\texttt{sdw}} h_{\mathrm{B}}} \sum_{\{y_i|x_i=\texttt{sdw}\}} \Phi\left(\frac{y - y_i}{h_{\mathrm{B}}}\right), \tag{7.8}$$

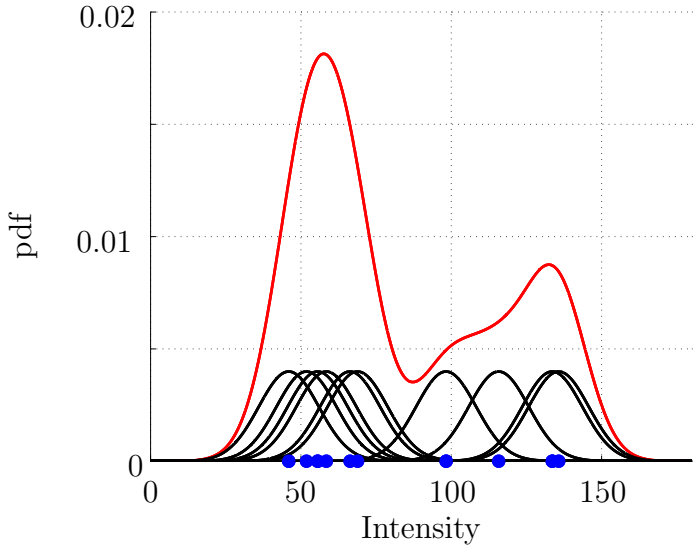


Figure 7.4: Gaussian kernel density estimation. The method is illustrated with 10 samples (blue points), their associated Gaussian curves (black), and the sum of them (red), representing the estimated pdf.

where Φ is some kernel, $N_{\texttt{sdw}}$ is the number of pixels assigned to the label `sdw` and h_{B} is a smoothing parameter called bandwidth [61]. Analogously, $\hat{p}_{\texttt{bkg}}(y)$ and $\hat{p}_{\texttt{hlt}}(y)$ are estimated. The Gaussian kernel,

$$\Phi\left(y\right) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}y^2\right\}, \tag{7.9}$$

has been used. Hence, the variance of the estimator is controlled through h_{B}. In this implementation h_{B} is optimized for Gaussian distributions. Fig. 7.4 illustrates a Gaussian density estimator. While the histogram groups the samples in bins, the kernel density estimator places a curve with the shape of the kernel at each sample position. The estimated pdf consists of the sum of all of them.

Fig. 7.3(b) shows the histogram and the non-parametric pdf of the SAS image in Fig. 7.1(a) as segmented in Fig. 7.1(b). The non-parametric pdfs approximate the histograms better than the Weibull and, mainly, the Rayleigh pdfs (in Fig. 7.3(a)), specially for the `hlt` label.

Since no parameter can be extracted to define the non-parametric pdfs, in this case, $\boldsymbol{\Omega}_y$ is assigned to the distributions themselves: $\boldsymbol{\Omega}_y = \{\hat{p}_{\texttt{sdw}}, \hat{p}_{\texttt{bkg}}, \hat{p}_{\texttt{hlt}}\}$.

All segmentation results presented in this thesis assume a non-parametric model for $p_{\mathbf{y}|\mathbf{x}}$.

7.2 Iterative Conditional Modes

The Iterative Conditional Modes (ICM) [106] is the most common method employed to approximate the global maximum of Eq. (7.1) by a local maximum. It constitutes a reasonably fast approach and produces fairly good results.

Assuming that some other method (e.g., the ICE in Sec. 7.2.1) provides estimates for the parameters $\mathbf{\Omega}_y$ (see Sec. 7.1.2) and $\mathbf{\Omega}_x$ (see Sec.7.1.1) and therewith, for $p_{y_i|x_i}$ and p_{x_i}, respectively, each pixel is considered in turns and is assigned a label according to the Maximum *A Posteriori* (MAP) criterion:

$$\hat{x}_i = \arg\max_{x_i}\{p_{x_i|y_i}\} = \arg\max_{x_i}\{p_{x_i} \cdot p_{y_i|x_i}\}, \quad x_i \in \{\texttt{bkg}, \texttt{sdw}, \texttt{hlt}\}. \tag{7.10}$$

This process is repeated until convergence. For the sonar image $\mathbf{y}$ in Fig. 7.1(a), the estimate of $\mathbf{x}$, $\hat{\mathbf{x}}$, is depicted in Fig. 7.1(b).

The ICE algorithm is described in the following. It requires an initialization of the segmentation, which is tackled in Sec. 7.2.2. The initialization has, indeed, a remarkable influence on the final ICM segmentation result. For this reason, special attention has been paid to it. Four different initialization schemes are compared. On the one hand, the two-step three-region implementation in [77] is used as a reference. On the other hand, thresholding, K-means and an enhanced initialization scheme proposed in this thesis are considered.

7.2.1 Iterative Conditional Estimation

The parameters $\mathbf{\Omega}_y$ and $\mathbf{\Omega}_x$ are estimated iteratively by means of the ICE algorithm [112]. After initializing $\mathbf{\Omega}_y^{[0]}$ and $\mathbf{\Omega}_x^{[0]}$ (see Sec. 7.2.2), the algorithm performs the following steps at each iteration l until convergence:

- **Step 1.** Use $\mathbf{\Omega}_x^{[l]}$ to calculate the Markovian probability p_{x_i} for the three labels: $p_{x_i}(x_i = \texttt{sdw})$, $p_{x_i}(x_i = \texttt{bkg})$ and $p_{x_i}(x_i = \texttt{hlt})$.
- **Step 2.** Use $\mathbf{\Omega}_y^{[l]}$ to calculate the likelihood functions for each pixel: $p_{y_i|\texttt{sdw}}$, $p_{y_i|\texttt{bkg}}$ and $p_{y_i|\texttt{hlt}}$.
- **Step 3.** Calculate the *a posteriori* probability for each label:

$$p_{x_i|y_i} = p_{x_i} \cdot p_{y_i|x_i}, \quad x_i = \texttt{sdw}, \texttt{bkg}, \texttt{hlt} \tag{7.11}$$

- **Step 4.** Use the Gibbs sampling algorithm to obtain N_{Gibbs} samples of the label field, $\hat{\mathbf{x}}_{(1)}, \cdots, \hat{\mathbf{x}}_{(N_{\text{Gibbs}})}$, according to $p_{\mathbf{x}|\mathbf{y}}$ and using $\mathbf{\Omega}_y^{[l]}$ and $\mathbf{\Omega}_x^{[l]}$.
- **Step 5.** For each sample $\hat{\mathbf{x}}_{(j)}$, estimate the parameter vectors $\hat{\mathbf{\Omega}}_x(\mathbf{x}_{(j)})$ and $\hat{\mathbf{\Omega}}_y(\mathbf{x}_{(j)})$, $1 \leq j \leq N_{\text{Gibbs}}$ (as described in Secs. 7.1.1 and 7.1.2, respectively).
- **Step 6.** Calculate $\mathbf{\Omega}_x^{[l+1]}$ as

$$\mathbf{\Omega}_x^{[l+1]} = \frac{1}{N_{\text{Gibbs}}} \sum_{j=1}^{N_{\text{Gibbs}}} \hat{\mathbf{\Omega}}_x(\mathbf{x}_{(j)}) \tag{7.12}$$

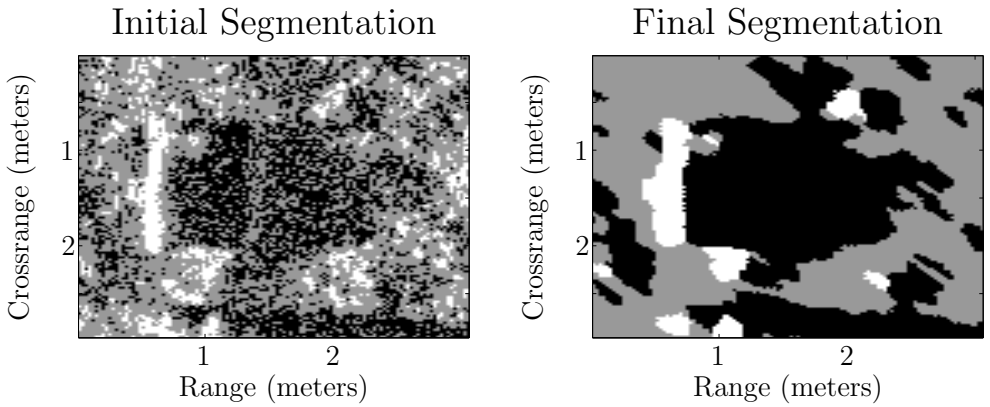
  and analogously, calculate $\mathbf{\Omega}_y^{[l+1]}$ from $\hat{\mathbf{\Omega}}_y(\mathbf{x}_{(j)}$, $1 \leq j \leq N_{\text{Gibbs}}$.

In this implementation, $N_{\text{Gibbs}} = 1$ has been chosen (see [119] for details).

7.2.2 Initialization

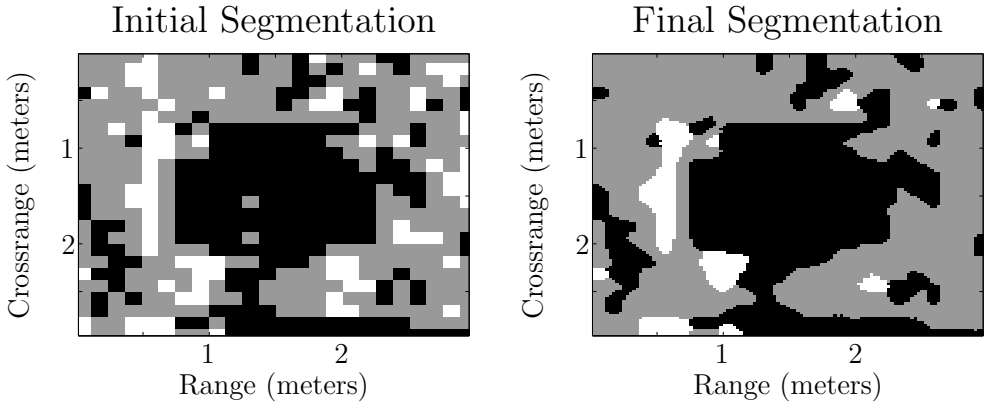
An initialization of the segmentation is required in order to estimate $\mathbf{\Omega}_y^{[0]}$ and $\mathbf{\Omega}_x^{[0]}$. It has a great effect on the ICE parameter estimation and therefore, on the ICM segmentation result. Furthermore, it influences the ICE convergence speed.

First in this section, two well-known segmentations algorithms, the thresholding and the K-means approaches are described. After, an enhanced initialization scheme, proposed in this thesis, is introduced. For comparison, the two-step three-region approach described in [77] has also been implemented and tested. In the first step, the [77] approach employs the ICM algorithm to segment the sonar image into two regions, `sdw` and `bkg`. In the second step, the highlight label `hlt` is initialized, and a three-region segmentation is iteratively estimated. The value of $\mathbf{\Omega}_x$ and $\mathbf{\Omega}_y$ are estimated only in the first step, staying constant during the second. The results provided by the four initialization schemes are compared in Sec. 7.2.2.4.



(a) Initialization by the thresholding method applied to $\mathbf{m} = \mathbf{y}$.

(b) Final segmentation result when the ICE is initialized as in Fig. 7.5(a).



(c) Initialization by the thresholding method applied to $\mathbf{m} = (\mu_1, \mu_2, \cdots, \mu_{N_\mathrm{W}})$

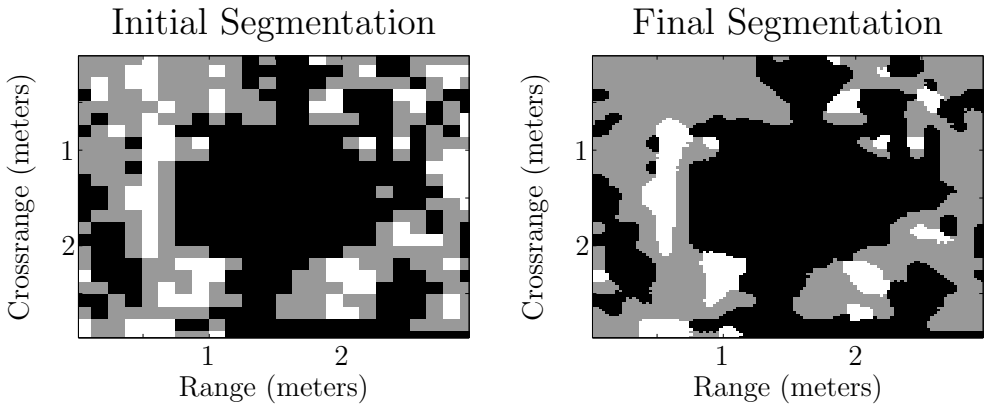
(d) Final segmentation result when the ICE algorithm is initialized as in Fig. 7.5(c).

Figure 7.5: Thresholding initialization applied to the SAS image in Fig. 7.1(a).

7.2.2.1 Thresholding

Thresholding is an intuitive segmentation technique [103]. If a set of data with N_W elements, $\mathbf{m} = \{m_1, m_2, \cdots, m_{N_\mathrm{W}}\}$, needs to be segmented into K groups, $K-1$ thresholds, z_k, $k = 1, \cdots, K-1$, are required. An element $m_j \in \mathbf{m}$ is assigned to region k if $z_{k-1} \leq m_j < z_k$. We assume $z_0 = \min\{\mathbf{m}\}$ and $z_K = \max\{\mathbf{m}\}$. The main disadvantage of the thresholding initialization scheme is that z_k, $1 \leq k \leq K-1$ must be chosen *ad hoc*. While a certain set of z_k leads to a successful segmentation result for a certain image, it may provide poor results for a different one.

It is possible to apply the thresholding segmentation directly to the sonar image, that is, $\mathbf{m} := \mathbf{y}$. The label `sdw` is assigned to the pixels of lower intensity: $x_i :=$ `sdw` $\forall i | z_0 \leq y_i < z_1$. Those in the interval $z_1 \leq y_i < z_2$ correspond to `bkg`, and the label `hlt` is reserved for the pixels in $z_2 \leq y_i < z_3$. The thresholds $z_1 = \min\{\mathbf{y}\} + 0.2 \cdot \Delta y$



(a) Initial segmentation provided by the K-means algorithm, where $K = 3$ classes have been used.

(b) Final segmentation result when the ICE algorithm is initialized as in Fig. 7.6(a).

Figure 7.6: K-means initialization applied to the SAS image in Fig. 7.1(a).

and $z_2 = \min\{\mathbf{y}\} + 0.5 \cdot \Delta y$, where $\Delta y = \max\{\mathbf{y}\} - \min\{\mathbf{y}\}$, lead to the initialization in Fig. 7.5(a) when applied to the SAS image in Fig. 7.1(a). It results in the final ICM segmentation illustrated in Fig. 7.5(b).

The following approach is however more convenient. A non-overlapping window is shifted along the image $\mathbf{y}$, and the mean μ_j of the pixel intensities is computed for each position. Applying the thresholding algorithm to $\mathbf{m} := (\mu_1, \mu_2, \cdots, \mu_{N_\mathrm{W}})$, where N_W is the total number of non-overlapping windows, results into the initial segmentation in Fig. 7.5(c), that leads to the final ICM segmentation result in Fig. 7.5(d). This implementation allows for shorter execution times. Many neighborhood configurations that appear in Fig. 7.5(a) are not realistic. Therefore, the estimation of $\mathbf{\Theta}_i$ (see Eq. (7.4)) in the first iterations of the ICE algorithm is poor and leads to either a poor final segmentation result or a higher number of iterations until convergence. For example, the result in Fig. 7.5(d) needs half as many iterations as Fig. 7.5(b).

An appropriate size for the window has to be chosen. In general, the smaller the window size, the more accurate is the initial segmentation result. However, it also encourages the appearance of small negligible groups of dark pixels being classified as `sdw`, or small groups of light pixels being classified as `hlt`. For the available data, a window of 5×5 pixels has demonstrated to produce satisfactory results.

7.2.2.2 K-means Algorithm

The K-means algorithm does not need *ad hoc* information, overcoming the main limitation of the thresholding initialization scheme. Given a set of data with N_W elements,

$\mathbf{m} = (m_1, m_2, \cdots, m_{N_\mathrm{W}})$, the K-means algorithm segments it into K regions according to the following procedure:

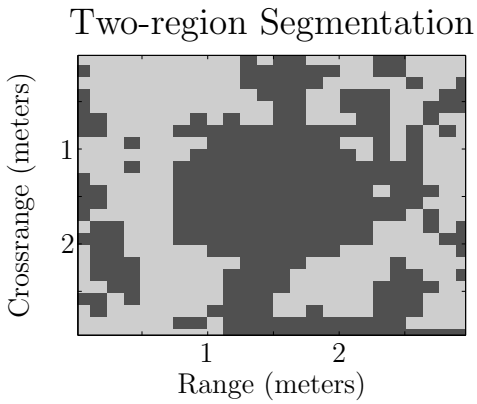
- **Step 1.** Randomly divide the elements of $\mathbf{m}$ into K clusters of the same number of elements.
- **Step 2.** Find the center of mass of each cluster, c_k, $k = 1, \cdots, K$.
- **Step 3.** Measure the distance d_{kj} between each data point m_j, $1 \leq j \leq N_\mathrm{W}$, and the center of mass of each cluster c_k, $1 \leq k \leq K$, according to some norm $||\cdot||$.
- **Step 4.** Assign each data point m_j, $1 \leq j \leq N_\mathrm{W}$, to the cluster that minimizes d_{kj},
$$k_j = \arg\min_k \{d_{kj}\}, \quad 1 \leq k \leq K. \tag{7.13}$$
- **Step 5.** Go to step 2.

This iterative process is repeated until convergence. The Euclidean distance has been chosen as norm. For the reasons stated in Sec. 7.2.2.1, the K-means algorithm is applied to the vector of means $\mathbf{m} = (\mu_1, \mu_2, \cdots, \mu_{N_\mathrm{W}})$, also with a window of size 5×5 pixels. The segmentation of the sonar image in Fig. 7.1(a) by the K-means algorithm with $K = 3$ is illustrated in Fig. 7.6(a). It leads to the final ICM segmentation in Fig. 7.6(b).

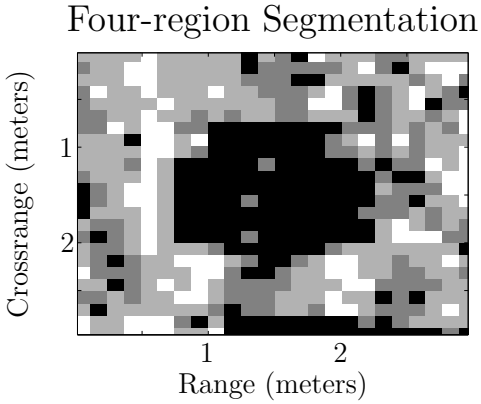
7.2.2.3 Enhanced Initialization Scheme

For the SAS example under consideration, the segmentation provided by the thresholding initialization in Fig. 7.5 is more accurate than the result corresponding to the K-means initialization in Fig. 7.6. However, the thresholding algorithm requires *ad hoc* values for z_k, which is a remarkable limitation when aiming for an ADAC system.

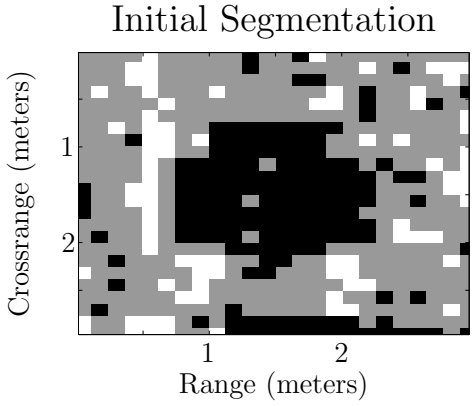
In this thesis, an enhanced initialization scheme is proposed. It provides an accurate initialization that yields high quality segmentation results also in scenarios where the statistics of some background areas are close to the statistics of the shadow regions. The steps of the initialization algorithm are summarized in the sequel:



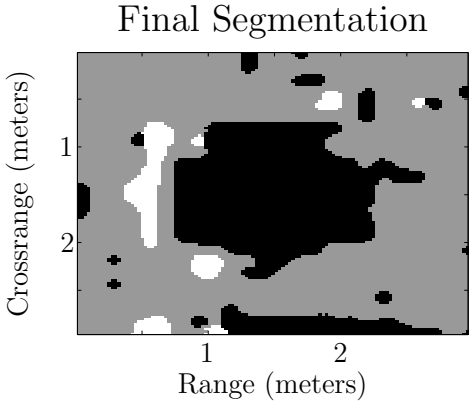
(a) Two-region segmentation result corresponding to the second step of the enhanced initialization scheme. The darkest pixels correspond to $\mathbf{m}_{\text{low}}$ while the lightest correspond to $\mathbf{m}_{\text{high}}$.



(b) Four-region segmentation (third step of the algorithm). The black pixels depict $\mathbf{m}^1_{\text{low}}$, the dark gray ones correspond to $\mathbf{m}^2_{\text{low}}$, the light gray pixels depict $\mathbf{m}^1_{\text{high}}$ and $\mathbf{m}^2_{\text{high}}$ is depicted in white.



(c) Segmentation produced by the enhanced algorithm. The `sdw` pixels are depicted in back, the dark and light gray regions in Fig. 7.7(b) are merged and assigned to `bkg`. The `hlt` label is depicted in white.



(d) Final segmentation result when the segmentation illustrated in Fig. 7.7(c) is used as initialization for the ICE.

Figure 7.7: Enhanced initialization applied to the SAS image in Fig. 7.1(a).

- **Step 1.** Use a non overlapping window to compute $\mathbf{m} = (\mu_1, \mu_2, \cdots, \mu_{N_{\text{W}}})$.
- **Step 2.** Sort the elements of $\mathbf{m}$ in ascending order and split the resulting vector into two vectors of the same length, $\mathbf{m}_{\text{low}}$ and $\mathbf{m}_{\text{high}}$.
- **Step 3.** Apply the K-means algorithm with $K = 2$ to both $\mathbf{m}_{\text{low}}$ and $\mathbf{m}_{\text{high}}$. Hence, $\mathbf{m}_{\text{low}}$ splits into $\mathbf{m}^1_{\text{low}}$ and $\mathbf{m}^2_{\text{low}}$ and $\mathbf{m}_{\text{high}}$ splits into $\mathbf{m}^1_{\text{high}}$ and $\mathbf{m}^2_{\text{high}}$.
- **Step 4.** Assign the label `sdw` to the elements in $\mathbf{m}^1_{\text{low}}$.
- **Step 5.** Merge $\mathbf{m}^2_{\text{low}}$ and $\mathbf{m}^1_{\text{high}}$ and assign them the label `bkg`.
- **Step 6.** Assign the label `hlt` to the elements in $\mathbf{m}^2_{\text{high}}$.

In order to illustrate the algorithm, it has been applied to the SAS image in Fig. 7.1(a). The initial two-region segmentation (step 2) is included in Fig. 7.7(a). At step 3, the K-means algorithm with $K = 2$ is applied to each of these regions, producing a four-region segmentation, as illustrated in Fig. 7.7(b). The labels `sdw` and `hlt` are assigned to the regions of lower and higher intensity, respectively, and the two-regions of intermediate intensity are merged together and assigned the label `bkg`. The resulting three-region segmentation, shown in Fig. 7.7(c), is employed as initialization for the ICE algorithm, and the ICM final segmentation result shown in Fig. 7.7(d) is obtained.

7.2.2.4 Comparison

The three initialization schemes presented above and the well-established two-step three-region approach described in [77] have been tested for the SAS1 database. A collection of snapshots and the corresponding ICM final segmentation results are illustrated in Fig. 7.8. Each row shows an SAS image and the segmentation result produced by all four approaches. The sonar images on the three top rows are segmented successfully by all schemes. These SAS images present a high contrast and very distinct shadow and background regions. The four last SAS images display a background with some dark areas close to the shadow intensity. They are poorly segmented by the two-step three-region approach, as well as by the ICM algorithm when either the thresholding or the K-means initialization are used. Only the enhanced initialization scheme provides a high quality segmentation in these cases. About 80 % of the elements in the SAS1 database are significantly better segmented by the enhanced initialization scheme, which has been adopted in the following.

7.3 Min-Cut/Max-Flow

Although the ICM algorithm provides satisfactory segmentation results for simple and moderately challenging scenarios, it has been observed that it fails in certain cases. Thus, sand ripples or backgrounds with parts almost as dark as the objects shadow can lead to poor segmentation results, as shown in Fig. 7.9, where the segmented shadow of the cylindrical object is much broader than the cylinder itself and has, moreover, a very irregular shape.

In [123] it is demonstrated that a min-cut/max-flow algorithm can be used to estimate the label field $\mathbf{x}$ of a MRF modeled image (see Sec. 7.1). Thus, a computationally

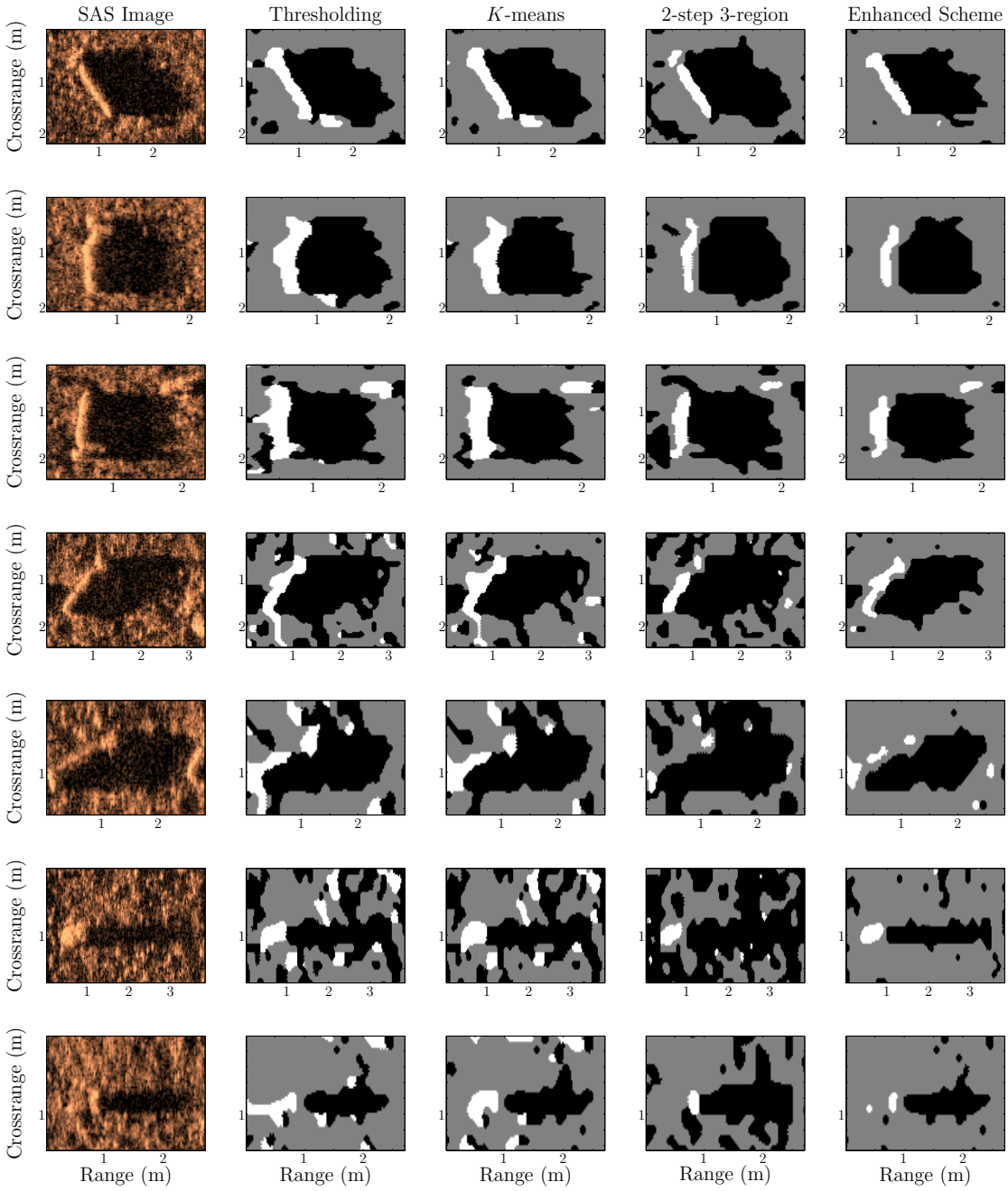


Figure 7.8: Segmentation result produced by the ICM algorithm when initialized by four different schemes. Each row corresponds to a different SAS image. On the first column, the sonar snapshot is shown. The second and third columns depict the segmentation results when thresholding and the K-means algorithms are employed for initialization, respectively. The forth column depicts the two-step three-region [77] segmentation results and the last one corresponds to the enhanced initialization scheme proposed in this thesis.

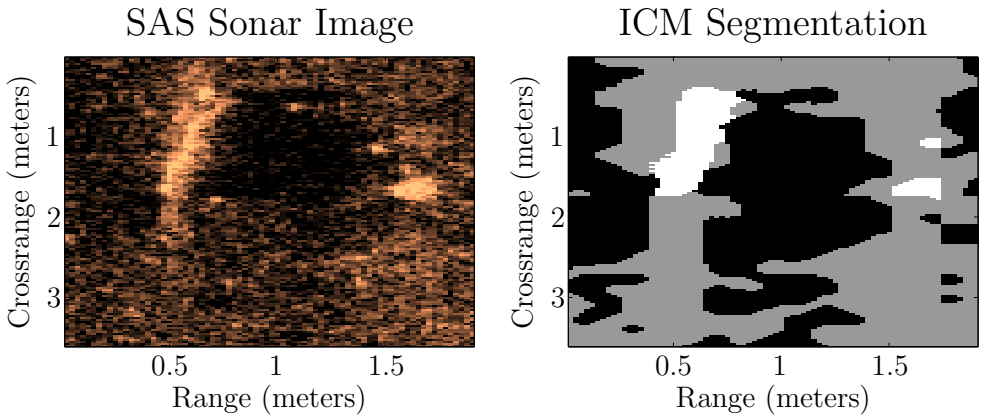


Figure 7.9: SAS image and ICM segmentation result.

efficient implementation of a graph cut algorithm [101] has been utilized to segment the sonar images. Using a graph representation for the image, a min-cut/max-flow algorithm splits it into two groups of pixels, one assigned to the shadow label `sdw` and the second to the background label `bkg`. The ICM result for the `hlt` label is in general satisfactory, and therefore is not regarded by the graph cut approach. To the best knowledge of the author, it is the first time that graph cut theory is applied for segmentation of sonar images.

First in this section an overview of graph theory is provided, with a focus on min-cut/max-flow algorithms. The modeling of the regional and boundary properties, related to $p_{\mathbf{y}|\mathbf{x}}$ and $p_{\mathbf{x}}$ (see Eq. (7.1)), are tackled in Sec. 7.3.2. In Sec. 7.3.3 the initialization issue is addressed. Finally, a parameter study is accomplished.

7.3.1 Graph Theory

A directed weighted graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ consists of a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$. An edge represents a connection between two ordered nodes, that is, $\mathcal{E} = \{\{i, i'\} | i, i' \in \mathcal{V}\}$. A function $g \colon \mathcal{E} \rightarrow \mathbb{R}^+$ assigns a positive real valued weight to each edge, denoted by $g_{\{i,i'\}}$. Since the graph is directed $g_{\{i,i'\}} \neq g_{\{i',i\}}$ [124].

Grid graphs are typically employed in computer vision to represent images, since the alignment of nodes in rows and columns is a natural representation of the image pixels $i \in \mathcal{L}$. A neighborhood system $\mathcal{M}$ has to be chosen to establish the edges configuration connecting the different pixel nodes. For the application at hand, the second order neighborhood system is considered (see Fig. 7.2).

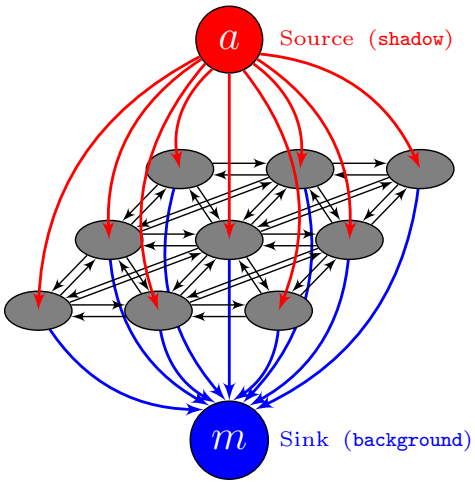


Figure 7.10: Graph representation of a 3×3 image with two terminals and a second order neighborhood system for the n-links.

For the purpose of image segmentation some extra nodes denoted as terminals are required. Each terminal corresponds to one of the possible pixel labels. For binary segmentation two terminals are required, the source a and the sink m. In a terminal graph, each pixel is connected not only to its eight neighbors, but also to the terminal nodes a and m. Hence, two kinds of edges are distinguished: n-links (neighboring links) are edges between two pixel nodes and t-links (terminal links) are edges between a pixel node and a terminal. Thus, the two-terminal graph $\mathcal{G}$ is defined by

$$\mathcal{V} = \mathcal{L} \cup \{a, m\} \tag{7.14}$$

$$\mathcal{E} = \mathcal{M} \cup \underbrace{\{\{i, a\}, \{i, m\} | i \in \mathcal{L}\}}_{\text{t-links}}. \tag{7.15}$$

A two-terminal graph representing a 3×3 image is depicted in Fig. 7.10.

7.3.1.1 Graph Cut

An a/m cut W (hereafter referred to only by cut) on a graph separates the set of nodes $\mathcal{V}$ into two disjoint subsets $\mathcal{S} \subset \mathcal{V}$ and $\mathcal{T} \subset \mathcal{V}$, $\mathcal{S} \cap \mathcal{T} = \emptyset$, such that the source $a \in \mathcal{S}$ and the sink $m \in \mathcal{T}$. In this application, $\mathcal{S}$ corresponds to the `sdw` label and $\mathcal{T}$ to the `bkg` label. A cut $W = \{\mathcal{S}, \mathcal{T}\}$ is a subset of $\mathcal{E}$ containing all edges $\{i, i'\}$ where $i \in \mathcal{S}$ and $i' \in \mathcal{T}$. The cost of a cut $|W|$ is defined as the sum of the weights of the edges in W, i.e., $|W| = \sum_{\{i,i'\} \in W} g_{\{i,i'\}}$. The minimum cut is defined as a cut on graph $\mathcal{G}$ that has minimum cost.

The min-cut/max-flow theorem states that for any directed linear graph the maximum flow value from a to m is equal to the cost of the minimum cut separating a and m [124]. In other words, finding the minimum cut of a graph is equivalent to finding its maximum flow.

In order to illustrate the concept of flow in a graph, let us interpret the directed graph as a network and the edges as pipes connecting the nodes. Each pipe has a certain capacity that corresponds to the weight of the edge $g_{\{i,i'\}}$. Now a flow $\phi(a,m)$ is pushed through the network leaving the source and arriving at the sink. According to the min-cut/max-flow theorem, $\phi_{\max} = |W|_{\min}$.

Before stating the conditions that define a finite flow ϕ in a network, let us denote all outgoing edges from node i and all incoming edges to node i by,

$$\begin{aligned} O(i) &= \{\{i,i'\} \in \mathcal{E} | i' \in \mathcal{V}\} \\ I(i) &= \{\{i',i\} \in \mathcal{E} | i' \in \mathcal{V}\}, \end{aligned} \tag{7.16}$$

respectively. The first condition is given by,

$$\sum_{i' \in O(i)} \phi(i,i') \; - \sum_{i' \in I(i)} \phi(i',i) = \begin{cases} \phi & \text{if } i = a \\ -\phi & \text{if } i = m \\ 0 & \text{otherwise} \end{cases} \tag{7.17}$$

which is comparable to Kirchhoff's current law. Assuming that outcoming flows are positive and incoming flows are negative, the sum of all outgoing and incoming flows must be zero for all nodes but the source and the sink. The flow emerging from the source, ϕ, is equal to the flow arriving at the sink. Secondly, capacities must be finite, i.e., $g_{\{i,i'\}} < \infty$. Finally, the flow within an edge cannot exceed its capacity, $\phi(i,i') \leq g_{\{i,i'\}}$.

7.3.1.2 Implementation

The min-cut/max-flow algorithm proposed in [101], which is broadly used in the literature, has been adopted. It is based on the augmenting path concept [124]. The algorithm works on a residual graph $\mathcal{G}_\phi$, which is initialized as $\mathcal{G}$. In each iteration, a path along non saturated edges from a to m is searched in $\mathcal{G}_\phi$. The smallest capacity along the path determines the maximum flow $\Delta\phi$ that can be pushed. The residual capacities of the edges along the augmented path are reduced by $\Delta\phi$, while the residual capacities of the reverse edges are increased by the same amount. The total flow from a to m is increased, $\phi = \phi + \Delta\phi$. The algorithm terminates when there is no more $a \rightarrow m$ possible paths.

7.3.2 Edge Weighting

Segmenting an image using graph theory is equivalent to finding the minimum cut of its associated graph. Therefore, the segmentation result is determined by the edge weights. There are two kinds of edges, the n-links and the t-links (see Sec. 7.3.1). The former link each pixel with its neighbors, while the latter link each pixel with the source and the sink. Hence, it is natural that the weights of the n-links account for the so-called boundary properties of the image (related to $p_{\mathbf{x}}$ in Eq. (7.1)) while the t-links depend on its regional properties ($p_{\mathbf{y}|\mathbf{x}}$ in Eq. (7.1)).

The cost of the label field $\mathbf{x}$ reads [125]:

$$E(\mathbf{x}) = \nu \cdot R(\mathbf{x}) \ + (1-\nu) \cdot B(\mathbf{x}) \tag{7.18}$$

where the coefficient $\nu \in [0,1]$ specifies the relative weighting of the regional property term $R(\mathbf{x})$ with respect to the boundary property term $B(\mathbf{x})$, and

$$\begin{aligned} R(\mathbf{x}) &= \sum_{i \in \mathcal{L}} R_i(x_i) \\ B(\mathbf{x}) &= \sum_{\{i,i'\} \in \mathcal{M}} B_{\{i,i'\}} \cdot (1 - \delta[x_i - x_{i'}]). \end{aligned} \tag{7.19}$$

Note that the boundary term associated to an edge, $B_{\{i,i'\}}$, contributes to $B(\mathbf{x})$ only if $x_i \neq x_{i'}$.

Let us describe how to assign the weights $g_{\{i,i'\}}$ to the edges so that the expression in Eq. (7.18) corresponds to the cost of the cut defined by a certain labeling $\mathbf{x}$, that is, $E(\mathbf{x}) = |W|$. Considering the definition of a cut on a two terminal graph (see Sec. 7.3.1.1), the following statements regarding a cut W are made:

- if $i \in \mathcal{S}$ then $\{i, m\} \in W$
- if $i \in \mathcal{T}$ then $\{i, a\} \in W$
- $\{i, i'\} \in W$ iff $i \in \mathcal{S}$ and $i' \in \mathcal{T}$.

For every node $i \in \mathcal{L}$, exactly one t-link is severed by the cut. If two neighboring pixels i and i' are labeled differently, the edge with its origin in $\mathcal{S}$ and destination in $\mathcal{T}$ is severed by the cut. Then, an assignment of weights to the edges of graph $\mathcal{G}$ according to Table 7.1 ensures a minimization of $E(\mathbf{x})$ by the minimum cut on $\mathcal{G}$.

edge	$g_{\{i,i'\}}$	link
$\{i, i'\}$	$(1-\nu) \cdot B_{i,i'}$	n-link
$\{i, a\}$	$\nu \cdot R_i(\texttt{bkg})$	t-link
$\{i, m\}$	$\nu \cdot R_i(\texttt{sdw})$	t-link

Table 7.1: Edge weighting

7.3.2.1 Regional Properties

The more likely a pixel i is to belong to a region, the lower the regional cost $R_i(x_i)$ of assigning the corresponding label to the pixel must be. Hence, it is reasonable to express the regional cost as a function U of the likelihood function $p_{y_i|x_i}$,

$$R_i(x_i) = U(p_{y_i|x_i}), \quad x_i = \{\texttt{sdw}, \texttt{bkg}\}. \tag{7.20}$$

Since $R_i(x_i)$ should decrease with increasing $p_{y_i|x_i}$, U must be a monotonically decreasing function. Two options have been considered:

$$U_1(z) = 1 - z \tag{7.21}$$

$$U_2(z) = -\ln(z). \tag{7.22}$$

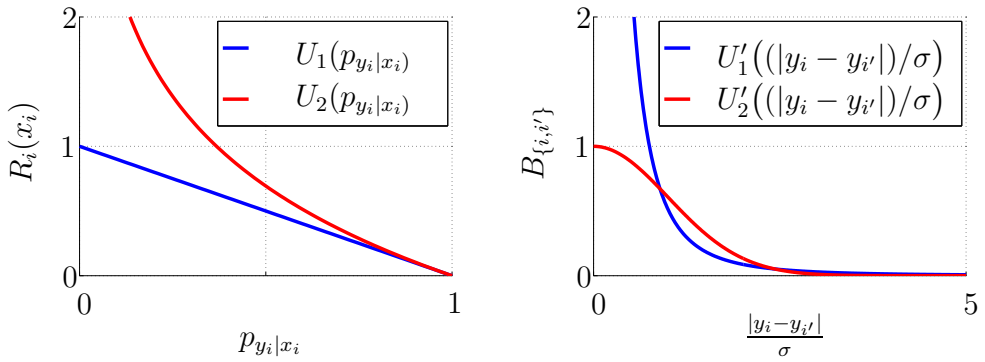
Fig. 7.11(a) shows U_1 and U_2 versus $p_{y_i|x_i}$. While U_1 and U_2 are similar for high values of $p_{y_i|x_i}$, they differ significantly for small $p_{y_i|x_i}$. The difference between the segmentation results that U_1 and U_2 produce is, however, negligible. This is due to the fact that edges with high weights do not get saturated and therefore do not determine the maximum flow. For the examples shown in Sec. 7.5, U_2 has been chosen.

The pdf of the two regions, $p_{y_i|\texttt{sdw}}$ and $p_{y_i|\texttt{bkg}}$, are estimated by maximization of the likelihood function of the seed pixels intensity (see Sec. 7.3.3). A Weibull distribution has been assumed for both regions (see Eq. (7.6)).

7.3.2.2 Boundary Properties

The boundary properties account for the fact that neighbor pixels with similar intensities should belong to the same region. Therefore, the boundary cost $B_{\{i,i'\}}$ can be defined as a function U' of the magnitude of the pixel intensity difference normalized by the standard deviation σ,

$$B_{\{i,i'\}} = U'\left(\frac{|y_i - y_{i'}|}{\sigma}\right). \tag{7.23}$$



(a) Regional properties (b) Boundary properties

Figure 7.11: Regional and Boundary properties. For U_1', $\omega_1 = \frac{1}{4}$ and $\omega_2 = 3$ have been chosen.

Two function families have been studied:

$$U_1'(z) = (\omega_1 + z)^{-\omega_2}, \quad \omega_2, \omega_1 \in \mathbb{Q}^+ \tag{7.24}$$

$$U_2'(z) = \exp\left(-\frac{z^2}{2}\right), \tag{7.25}$$

where ω_2 and ω_1 need to be chosen. An example of each function family is depicted in Fig. 7.11(b). Both functions are similar when $|y_i - y_{i'}| > \sigma$, but differ greatly otherwise. Again, the segmentation results that both functions produce are almost identical, since only the high weights differ and those do not influence the minimum cut. For the examples shown in Sec. 7.5, U_2' has been used.

Finally, let us present a modification of U' that, taking into account *a priori* information about the sonar images, allows for an improvement of the segmentation results. It is known that the intensity of the shadow regions is lower than that of the background. Thus, exploiting the fact that a cut always severs edges from the source to the sink (never from the sink to the source), the following modification to $B_{\{i,i'\}}$ is introduced:

$$B_{\{i,i'\}} = \begin{cases} U'\left(\frac{|y_i - y_{i'}|}{\sigma}\right) & \text{if } y_i < y_i \\ U'_{\max} & \text{if } y_i \geq y_{i'}. \end{cases} \tag{7.26}$$

A high value for $U'_{\max}$, e.g., $U'_{\max} = U'(0)$, discourages cuts where the `sdw` pixels have higher intensity than the `bkg` pixels.

7.3.3 Initialization: Seeds

It is possible to fix the label of a group of pixels, the so-called seeds. The subsets $\mathcal{O} \subset \mathcal{L}$ and $\mathcal{B} \subset \mathcal{L}$, $\mathcal{O} \cap \mathcal{B} = \emptyset$, denote the sets of seeds that are initially labeled as

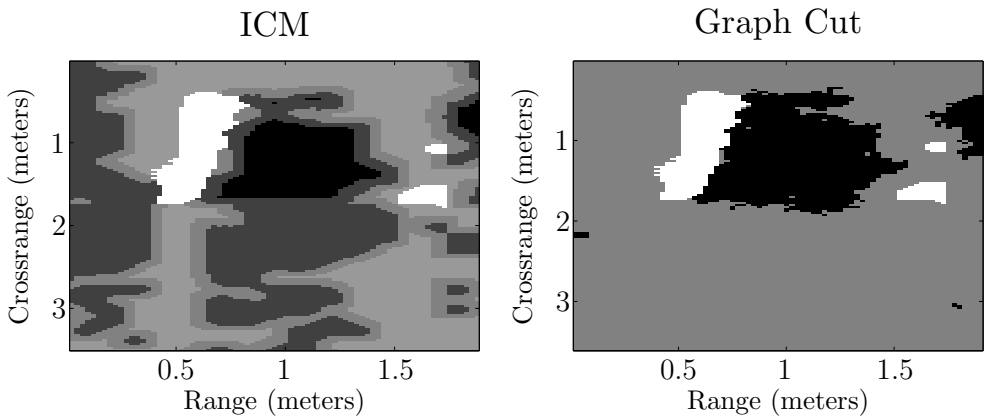


Figure 7.12: ICM (left) and graph cut (right) segmentation results with $h = 0.2$ and $\nu = 0.1$ for the sonar image in Fig. 7.9. The seeds for the graph cut algorithm initialization are highlighted in the ICM image.

`sdw` and `bkg`, respectively. They influence the labeling of the adjacent pixels via the boundary properties.

In this application the seeds are chosen after the ICM segmentation result. First, a rectangular structuring element is employed to morphologically erode [103] the shadow region. Its dimensions are proportional to those of the ICM shadow, that is, if the smallest rectangle that completely contains the ICM segmented shadow has size N_{C} and N_{R} in crossrange and range directions, respectively, then the structuring element has dimensions $h \cdot N_{\mathrm{C}} \times h \cdot N_{\mathrm{R}}$, with $h \in (0, 1)$. The pixels that remain and that, moreover, lie directly to the right of a highlight region (for a given crossrange), are added to the $\mathcal{O}$ set of seeds. After, the `bkg` region is eroded and the remaining pixels are considered as $\mathcal{B}$ seeds. The greater h is, the less pixels are assigned to the seed sets, that is, the graph cut segmentation is less influenced by the ICM result. Fig. 7.12 includes the ICM and the graph cut segmentation results for the SAS image in Fig. 7.9. The seeds that stem from the ICM segmentation with $h = 0.2$ are highlighted.

The seed pixels cannot change label during the max-flow search. According to [125] this is achieved by setting:

- if $i \in \mathcal{B}$, $g_{\{i,a\}} = 0$ and $g_{\{i,m\}} = 1 + \max_{i \in \mathcal{L}} \sum_{i':\{i,i'\} \in \mathcal{M}} B_{\{i,i'\}}$
- if $i \in \mathcal{O}$, $g_{\{i,m\}} = 1 + \max_{i \in \mathcal{L}} \sum_{i':\{i,i'\} \in \mathcal{M}} B_{\{i,i'\}}$ and $g_{\{i,a\}} = 0$.

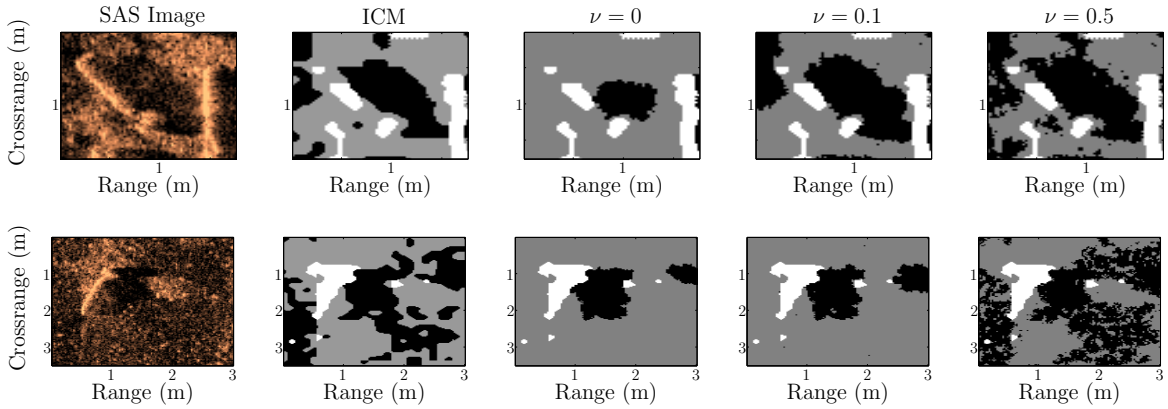


Figure 7.13: Study on ν. The ICM segmentation result of two SAS images is compared with the graph cut results stemming from $\nu = \{0, 0.1, 0.5\}$ assuming $h = 0.1$. The first example presents a poor segmentation for $\nu = 0$ (result too strongly determined by the initialization). The second example illustrates the degradation of the results due to a too high ν.

7.3.4 Parameter Study

An empirical study has determined suitable values for the parameters ν (see Eq. (7.18)) and h (see Sec. 7.3.3). In Fig. 7.13 the segmentation of two SAS images is shown, assuming values 0, 0.1 and 0.5 for ν. A value $h = 0.1$ has been chosen. If $\nu = 0$ only the boundary properties are considered. With $\nu = 0.5$ both regional and boundary properties have the same weight. While the latter configuration is sensitive to noise, the former is too strongly determined by the initialization. A good trade-off is $\nu = 0.1$.

Fig. 7.14 shows the segmentation that corresponds to $h = \{0.05, 0.1, 0.15\}$. Low values of h imply that most of the pixels are used as seeds. Hence, the graph cut segmentation is too much influenced by the ICM result and does not add any significant value. On the other hand, if h is too high, too few seed pixels are considered to estimate the pdf for the regional weights (see Sec. 7.3.2.1), which might result in a poor performance. A good compromise is $h = 0.1$.

7.4 Active Contours

Besides pixel labeling algorithms such as the two above, a second approach is often adopted. It consists of indicating the edge between different regions -object contours- by a line, and it is naturally linked to edge detection algorithms such as Active Contours (AC) or the level set method [114].

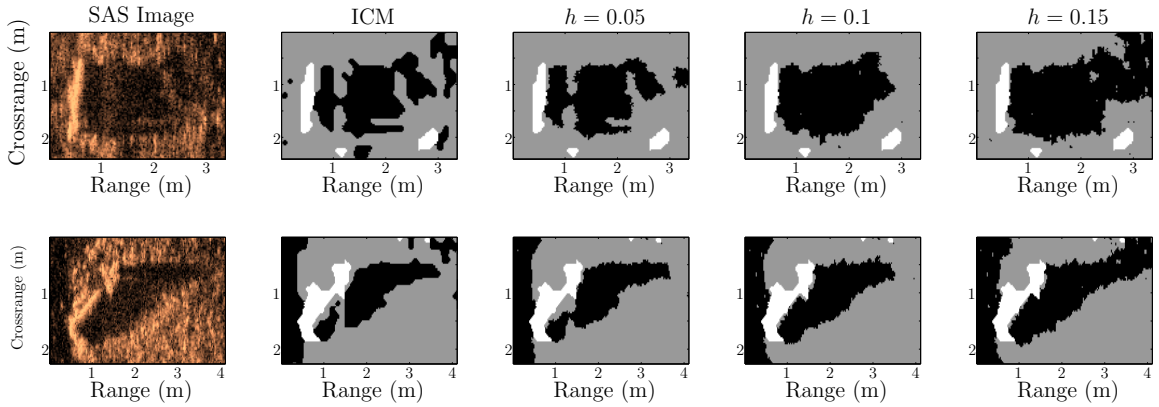


Figure 7.14: Study on h. The ICM segmentation result of two SAS images is compared with the graph cut results for $h = \{0.05, 0.1, 0.15\}$ and $\nu = 0.1$. The graph cut result for the first examples is very close to the ICM solution for $h = 0.05$. For higher values of h, the result differs and is more convenient for further classification purposes. Both examples show a poor segmentation for high h values ($h = 0.15$).

Typically used for medical imaging applications [126], the AC algorithm [113] has been successfully applied to sidescan sonar image segmentation [75, 77]. Unlike the Markovian segmentation approaches in Secs. 7.2 and 7.3, the AC algorithm does not assume any *a priori* probability of the regions, only the intensity values of the pixels influence the segmentation result.

An active contour (or statistical snake) $\mathbf{b} = \{\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_V\}^{\mathrm{T}}$, is a closed curve defined by a list of V nodes that has the ability to evolve in order to match the contour of an object present in an image [113]. Each node, expressed in Cartesian coordinates $\mathbf{b}_j = \{u_j, v_j\}$, corresponds to a pixel i in $\mathcal{L}$. The abscissa u is parallel to the range direction and the ordinate v indicates the crossrange position. The image of interest is therefore divided into two regions, the target region `tgt` inside $\mathbf{b}$ and the background region `bkg` outside $\mathbf{b}$. For segmentation of sonar images, the former corresponds to either the shadow or the highlight of an object.

7.4.1 Cost Function

The objective of the AC algorithm is to deform $\mathbf{b}$ in such a way that a given cost function $F(\mathbf{b})$ is minimized. Originally, the AC algorithm was based on the gradient so that $F(\mathbf{b})$ is minimum when $\mathbf{b}$ coincides with an edge between two regions. However, this approach performs poorly for noisy images and is very sensitive to initialization. Another approach to the problem consists of using parametric shape templates [127,

128]. Such templates are in general too stiff for an application like SAS, where the target regions present a broad variability.

A polygonal active contour that minimizes a cost function based on the likelihood function of the image is considered here [129]. Assuming independence among pixels, the cost function associated with a position of the active contour $\mathbf{b}$ is defined as the negative log-likelihood function of $\mathbf{y}$

$$F(\mathbf{b}) = -\ln\Big(\prod_{i\in\mathcal{L}} p_{y_i|x_i}\Big) = -\ln\Big(\prod_{x_i|i\in\mathtt{bkg}} p_{\mathtt{bkg}}(y_i)\cdot\prod_{x_i|i\in\mathtt{tgt}} p_{\mathtt{tgt}}(y_i)\Big), \tag{7.27}$$

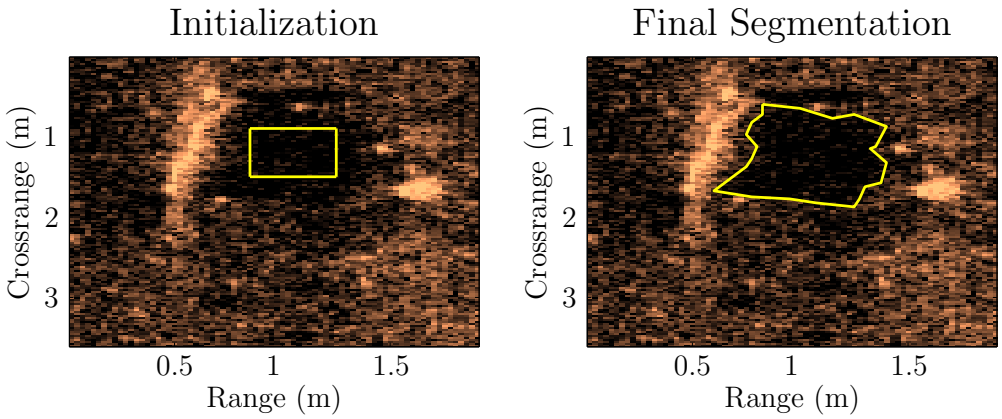
where $p_{\mathtt{tgt}}$ and $p_{\mathtt{bkg}}$ refer to the pdfs of the target and background regions, respectively and need to be estimated. An efficient implementation of Eq. (7.27) is possible for certain parametric models of the pdfs [129], among them the Rayleigh distribution. Both the Rayleigh and the Weibull distributions are suitable for sonar images (see Sec. 7.1.2). Since no efficient implementation of the AC algorithm is available under the Weibull model, the former has been selected. Thus, $p_{\mathtt{tgt}}$ and $p_{\mathtt{bkg}}$ are modeled by the Rayleigh distributions $\mathcal{R}(\alpha_{\mathtt{tgt}})$ and $\mathcal{R}(\alpha_{\mathtt{bkg}})$, respectively. The parameters $\alpha_{\mathtt{tgt}}$ and $\alpha_{\mathtt{bkg}}$ are estimated according to Eq. (7.7).

7.4.2 Initialization

The initialization of the AC algorithm is crucial. If $\mathbf{b}$ is initialized without comprising at least a piece of the target, the contour will diverge. The ICM segmentation result has been used to initialize the AC. If the AC initialization is identical to the ICM final segmentation, the final AC and ICM segmentations are practically identical and therefore, the AC segmentation result does not provide new information. To avoid this, the initialization of $\mathbf{b}$ is based on the ICM result but is, at the same time, significantly different from it: a rectangle centered in the center of mass of the ICM segmented region has been adopted. If the target corresponds to the shadow region, only the shadow pixels that, for a given crossrange, lie directly to the right of the associated ICM segmented highlight are considered in the calculation of the center of mass (analogously to the min-cut/max-flow initialization in Sec. 7.3.3). In Fig. 7.15(a), the initialization of the AC algorithm after the ICM segmentation in Fig. 7.9 is shown.

7.4.3 Implementation

The active contour is hence initialized with only four nodes. Subsequently, two main task are to be accomplished: increase of the number of nodes and optimization of their



(a) Initialization of the AC for the segmentation of the shadow region.

(b) Final AC segmentation result.

Figure 7.15: Active Contours algorithm.

position. The number of times that nodes are added to the initial 4-node contour $\mathbf{b}^{[0]}$ is counted by the index l. When the optimal position for $\mathbf{b}^{[l]}$ is achieved, $\mathbf{b}^{[l+1]}$ is initialized: if two consecutive nodes are further apart than a certain distance $d_{\max}$, V' nodes are inserted between them. Good results have been found for $d_{\max} = 6$ pixels and $V' = 1$ or 2 pixels.

The index l' is set to 0 every time that l increases and counts the number of iterations that are performed for a fixed amount of nodes. At each iteration, a single node $\mathbf{b}_j$, $1 \leq j \leq V$ of the current best active contour, $\mathbf{b}^{[l]}$, is shifted by a random distance, resulting in $\hat{\mathbf{b}}^{[l]}$. If $F(\hat{\mathbf{b}}^{[l]}) < F_{\min}$, $\mathbf{b}^{[l]} := \hat{\mathbf{b}}^{[l]}$ is adopted. Furthermore, the so-called crossing test [129] is performed to prevent intersections between the different segments of $\mathbf{b}$.

A common problem of the AC algorithm is that $\mathbf{b}$ may converge to local minima, leading to a poor segmentation. If a number of iterations $l_{\max}$ elapses after the last time that new nodes were added and no update was produced in $\mathbf{b}^{[l]}$, the algorithm might be stuck in a local minimum. Thus, an alternative active contour, $\tilde{\mathbf{b}}^{[l]}$, is calculated by shifting all nodes in $\mathbf{b}^{[l]}$ by Δ pixels. If the final cost function $F(\tilde{\mathbf{b}}^{[l]})$ is smaller than the original cost function $F(\mathbf{b}^{[l]})$, then $\mathbf{b}^{[l]} := \tilde{\mathbf{b}}^{[l]}$. This modification of the standard AC algorithm has demonstrated to improve its performance in more than 35 % of the cases. Good results have been obtained for $\Delta = 1$ pixel and $l_{\max} = 200$ iterations.

Hence, the implementation of the algorithm reads as follows. After initializing $\mathbf{b}^{[0]}$ with 4 nodes, estimating $p_{\texttt{bkg}}$ and $p_{\texttt{tgt}}$, and calculating $F_{\min} := F(\mathbf{b}^{[0]})$, the following iterative process is repeated while nodes in $\mathbf{b}^{[l]}$ are sparse:

- **Step 1.** $l' := 0$
- **Step 2.** While $l' < l_{\max}$
  - $l := l' + 1$
  - Choose a node $\mathbf{b}_j \in \mathbf{b}^{[l]}$ (randomly or sequentially) and shift it by a random distance: $\hat{\mathbf{b}}^{[l]}$
  - Estimate $p_{\mathtt{bkg}}$ and $p_{\mathtt{tgt}}$ and calculate $F(\hat{\mathbf{b}}^{[l]})$
  - If $F(\hat{\mathbf{b}}^{[l]}) < F_{\min}$, $\mathbf{b}^{[l]} := \hat{\mathbf{b}}^{[l]}$, $F_{\min} := F(\hat{\mathbf{b}}^{[l]})$
- **Step 3.** If $\mathbf{b}^{[l]} = \mathbf{b}^{[l-1]}$ (local minimum?), $\tilde{\mathbf{b}}^{[l]} := \mathbf{b}^{[l]} + \Delta$, else $\mathbf{b}^{[l+1]} :=$ add nodes $(\mathbf{b}^{[l]})$

The AC segmentation result of the example SAS image is shown in Fig. 7.15(b).

7.5 Results

This section presents a comparison of the results provided by the three segmentation algorithms: ICM, min-cut/max-flow and AC. A more meaningful comparison of the algorithms, based on the classification results that they provide, is available in Sec. 9.4.

The three segmentation algorithms employ sets of parameters that must be chosen in order to optimize the results. These parameters have been presented as the algorithms were introduced in Secs. 7.2 to 7.4. All ICM parameters are estimated from the image, e.g., $\mathbf{\Omega}_x$ or the likelihood function. On the contrary, both the min-cut/max-flow and AC algorithms require parameters that must be chosen heuristically. Values have been suggested in the corresponding sections. For the sake of clarity, they are summarized in Table 7.2. While the min-cut/max-flow algorithm is sensitive to changes in the parameter values (see Sec. 7.3.4), the AC algorithm is rather insensitive. For instance, similar segmentation results have been obtained for the parameter values listed in Table 7.2 and for the set: $d_{\max} = 4$ pixels, $V' = 2$ pixels, $\Delta = 2$ pixels and $l_{\max} = 150$ iterations.

SAS images are typically big (several thousands of square meters) and their statistical properties might vary within a single image. Furthermore, the ICM algorithm is computationally more efficient when applied to smaller images (see Sec. 7.6). Therefore, it

	Parameter	Value
min-cut/ max-flow	ν	0.1
	h	0.1
AC	$d_{\max}$	6 pixels
	V'	1 pixel
	Δ	1 pixel
	$l_{\max}$	200 iterations

Table 7.2: Parameter values

is advantageous to split each SAS image into several sub-images, so that a more accurate and efficient segmentation can be performed. The scheme that has been adopted to obtain the segmentation results presented in this thesis reads as follows:

- **Step 1.** Divide the SAS image $\mathbf{Y}$ into smaller sub-images $\mathbf{Y}_j$, e.g., of size 5×5 meters, $1 \leq j \leq N_{\text{sub}}$
- **Step 2.** Apply the ICM algorithm to each $\mathbf{Y}_j$ to obtain the segmented sub-image $\mathbf{X}_j^{\text{ICM}}$, $1 \leq j \leq N_{\text{sub}}$
- **Step 3.** Merge together all $\mathbf{X}_j^{\text{ICM}}$, $1 \leq j \leq N_{\text{sub}}$, to form $\mathbf{X}^{\text{ICM}}$
- **Step 4.** Scan $\mathbf{X}^{\text{ICM}}$ to obtain the object database:
  - Create a database entry s for each shadow region in $\mathbf{X}^{\text{ICM}}$, $1 \leq s \leq S$
  - If the shadow s has highlights to its left, associate them to s
  - Apply the AC algorithm to the shadow and all highlights associated with observation s, $1 \leq s \leq S$
  - Apply the min-cut/max-flow algorithm to each shadow s, $1 \leq s \leq S$

The number of sub-images is denoted by N_{sub}. Thus, each element s of the database has associated a snapshot of $\mathbf{Y}$ around the object of interest, and the snapshot segmentation result obtained by the three algorithms. Note that the database has an entry for each shadow but not for each highlight. Indeed, if no shadow is to the right of a highlight, the highlight is discarded. This is in agreement with the great variability that the highlight regions typically present in sonar imagery, in contrast with the more predictable shadow regions. Thus, objects with a certain orientation with respect to the sonar incident wave produce no return wave, which results in no highlight. The shadow of the object, however, is always visible.

In the case that more than one highlight lie to the left of a shadow, one of them has to be selected. The following approach has been adopted. For each highlight candidate, its area $\gamma_{\mathtt{hlt}}$ and the rate between its width and the shadow width along the crossrange direction, $r_{\mathtt{sdw,hlt}}$, are calculated. The highlight that obtains the highest value for the expression

$$\frac{1}{2} \cdot \frac{\gamma_{\mathtt{hlt}}}{\gamma^*_{\mathtt{hlt}}} + \frac{1}{2} \cdot r_{\mathtt{sdw,hlt}}, \tag{7.28}$$

is selected, where $\gamma^*_{\mathtt{hlt}}$ is the area of the biggest highlight candidate.

Figs. 7.16, 7.17 and 7.18 illustrate the segmentation results for different cylindrical, spherical and clutter objects from the SAS1 database, respectively. The first column shows the SAS image and, superimposed, the ICM segmentation result. The min-cut/max-flow and AC results are included in the second and the third column, respectively. Note that, in order to facilitate the comparison of the results, the ICM and min-cut/max-flow results are represented in the same format as the AC results, that is, the contours of the regions are depicted instead of the pixel labels.

The examples on the first three rows of Fig. 7.16 show high quality images. The ICM segmentation result is good and both the AC and graph cut segmentations are almost identical. A dark background area is segmented together with the shadow in the forth example by the ICM algorithm. The initialization of the graph cut algorithm with seeds that lie to the right of the highlight region allows for distinguishing the shadow of the object. Analogously, the AC algorithm, whose initialization also regards the ICM highlight segmentation, provides a good result as well. Presumably, this object will be correctly classified if either the min-cut/max-flow or the AC segmentations are employed to extract the shadow shape features. By contrast, it will be most probably classified as clutter if the ICM segmentation result is used.

Most spherical objects in Fig. 7.17 lie on uneven seabed regions with sand ripples. The ICM algorithm segments some of the sand ripples shadows together with the shadow of the objects. Thanks to their initialization, the graph cut and AC algorithms avoid this problem in most cases. Again, the min-cut/max-flow and AC segmentation results seem more likely to provide optimal classification results, since the shape of the segmented regions characterize the objects better. On the other hand, a set of features presented in Chapter 8 aims for a correct description of the spheres shadow shape when this kind of poor segmentation scenarios occur.

Fig. 7.18 depicts snapshots of uneven seabed areas that, when segmented, cause clutter objects. The intensities of the background and shadow regions are different, but this difference is not as pronounced as when prominent objects lie on the seabed. Therefore,

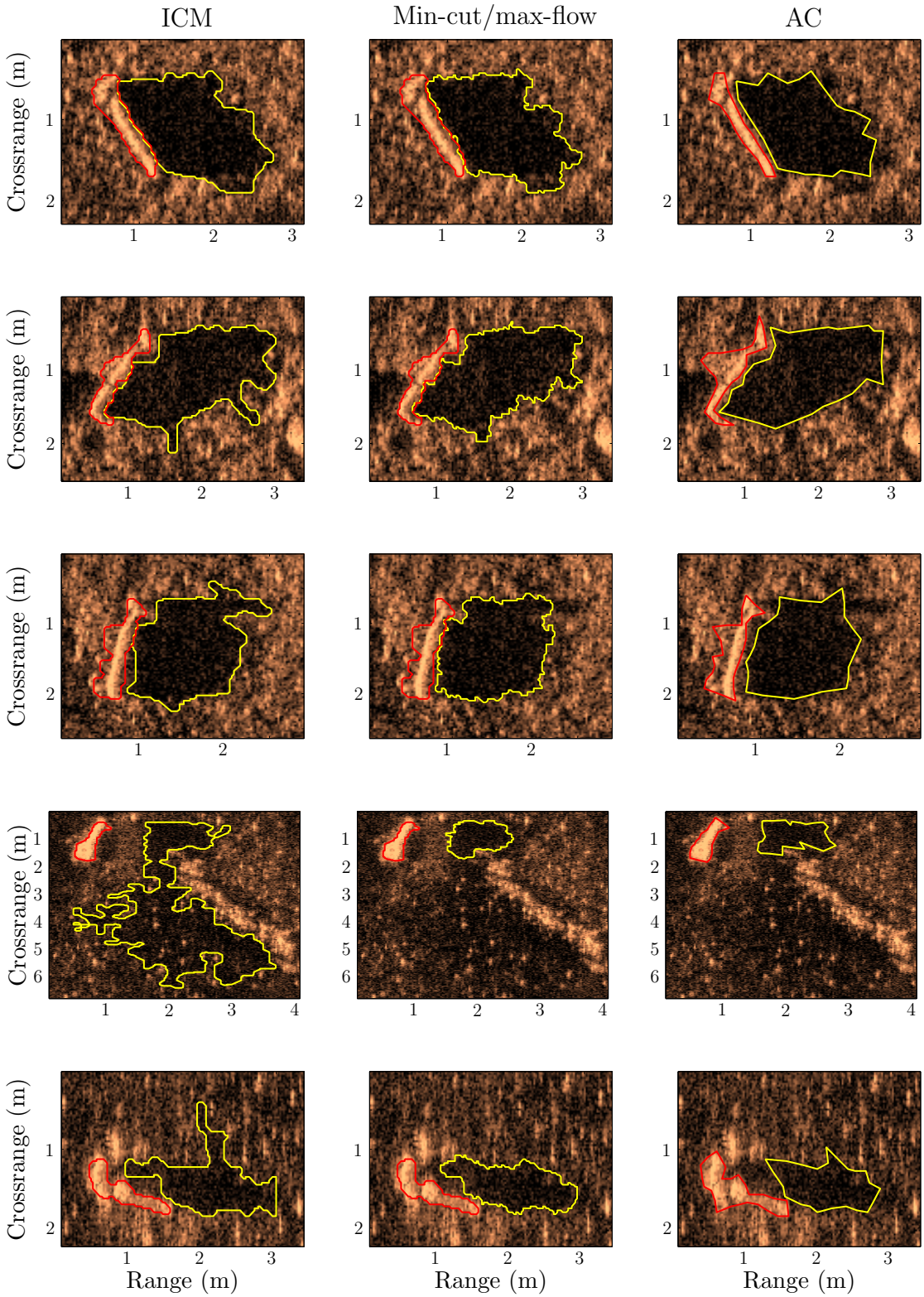


Figure 7.16: Comparison of segmentation performance for five cylindrical mines. The first column shows different snapshots of SAS images and, superimposed, the ICM segmentation result (red line for highlights and yellow line for shadows). The second and third column show the min-cut/max-flow and AC results, respectively.

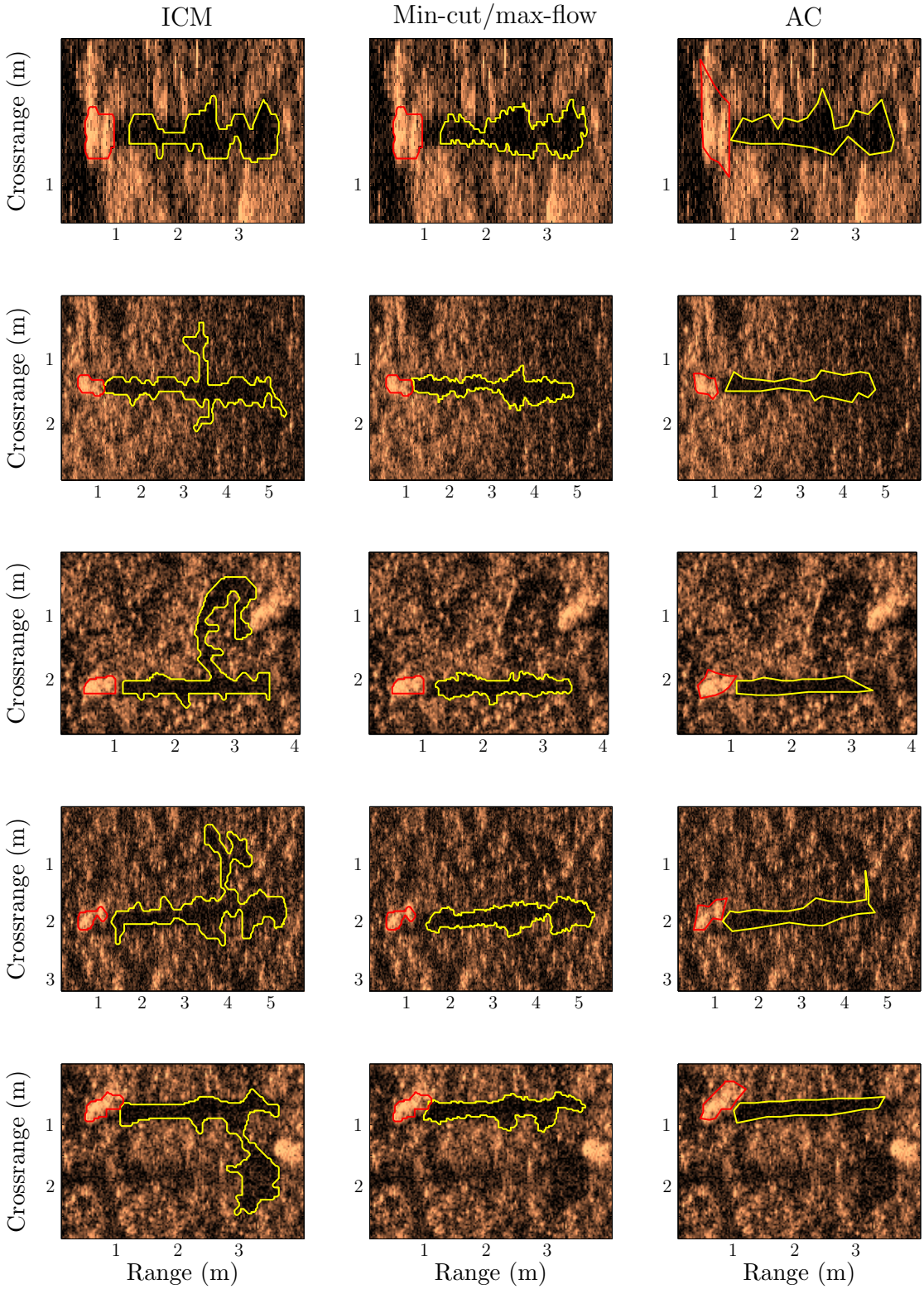


Figure 7.17: Comparison of segmentation performance for five spherical man made objects. The first column shows different snapshots of SAS images and, superimposed, the ICM segmentation result (red line for highlights and yellow line for shadows). The second and third column show the min-cut/max-flow and AC results, respectively.

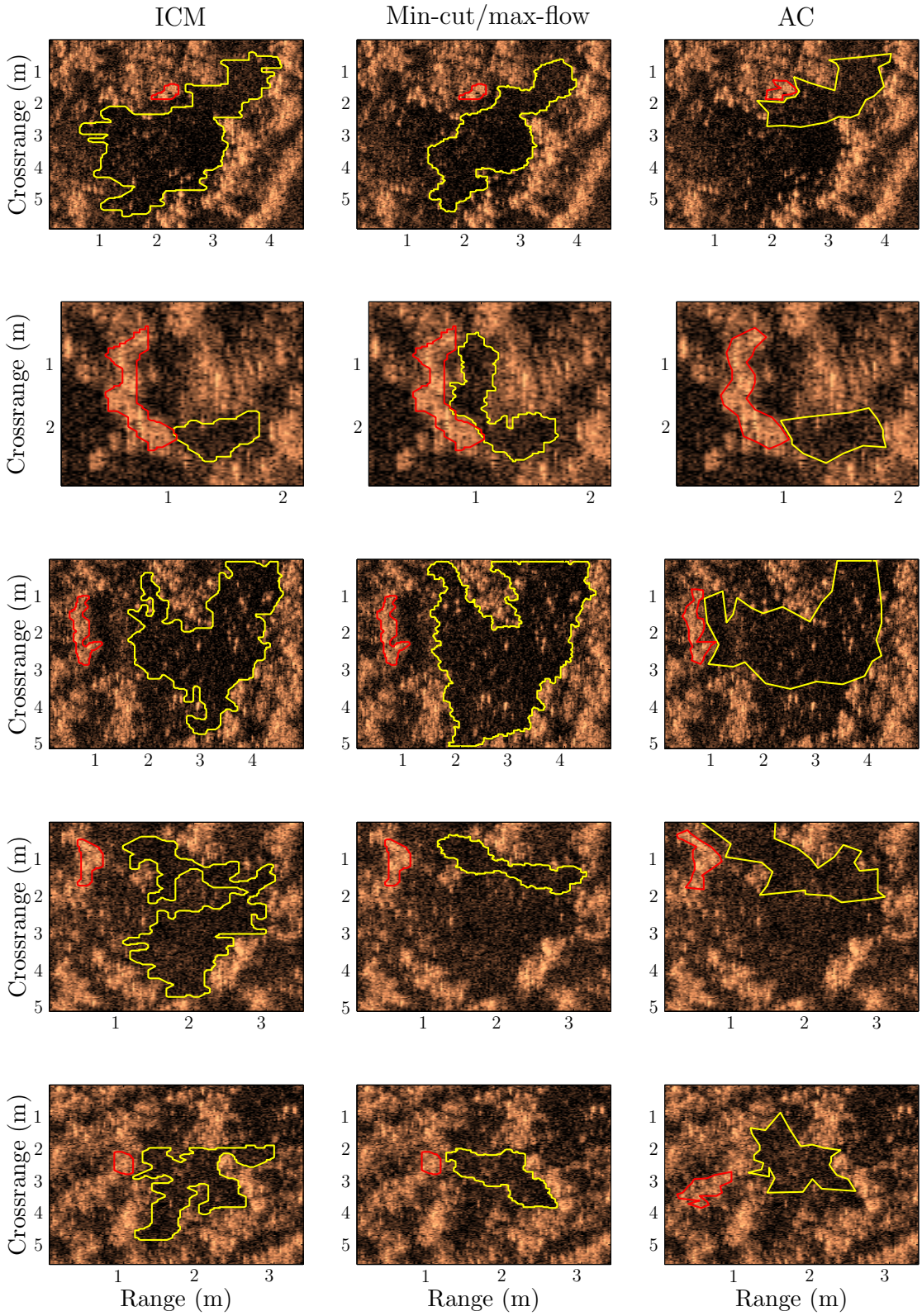
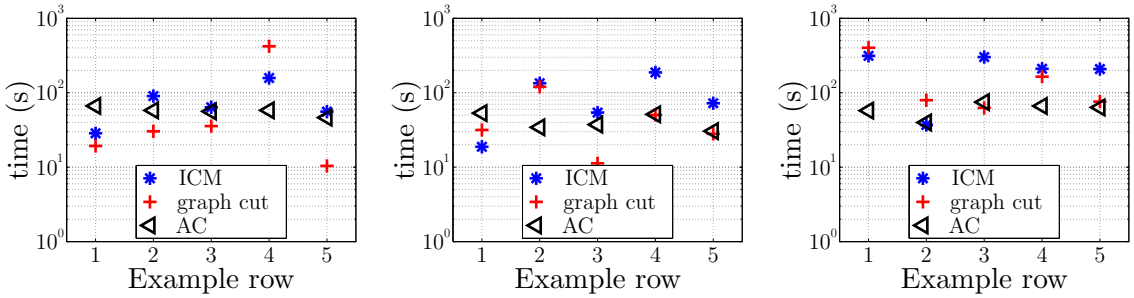


Figure 7.18: Comparison of segmentation performance for five regions that produce clutter. The first column shows different snapshots of SAS images and, superimposed, the ICM segmentation result (red line for highlights and yellow line for shadows). The second and third column show the min-cut/max-flow and AC results, respectively.



(a) Cylinders in Fig. 7.16. (b) Spheres in Fig. 7.17. (c) Clutter objects in Fig. 7.18.

Figure 7.19: Computational cost of the segmentation.

the segmentation results provided by the three algorithms diverge the most for clutter objects (see, e.g., the two last examples in Fig. 7.18). This is exploited in Sec. 8.4 in order to define a feature that measures the resemblance of the segmented regions provided by the different algorithms.

False alarms are prone to be produced by clutter objects whose segmentation resembles the shape of a mine. For example, the min-cut/max-flow segmentation of the forth example in Fig. 7.18 is somehow similar to the segmented shape of a spherical object and it is therefore likely to be classified as such.

7.6 Computational Cost

In order to study the computational cost of the three segmentation algorithms, the SAS image snapshots in Figs. 7.16, 7.17 and 7.18 have been employed. The computational time required by each algorithm is illustrated in Fig. 7.19. For the AC algorithm, the times employed to find the contours of both the shadow and highlight regions have been added up. A computer equipped with an Intel i5 4 core 2.8 GHz processor has been employed. The programs are written in Matlab [130].

All three algorithms are iterative and hence, the computational time depends on the number of iterations until convergence. The ICM algorithm requires in general less than 30 iterations. By contrast, both the graph cut and AC algorithms iterate several hundreds times before convergence is reached. The computational cost of one iteration though, is greater for the ICM algorithm, and the total times required by the three algorithms have the same order of magnitude. However, the initialization of the AC and min-cut/max-flow approaches is based on the ICM result. Hence, in practice, both methods are twice as slow as the ICM algorithm.

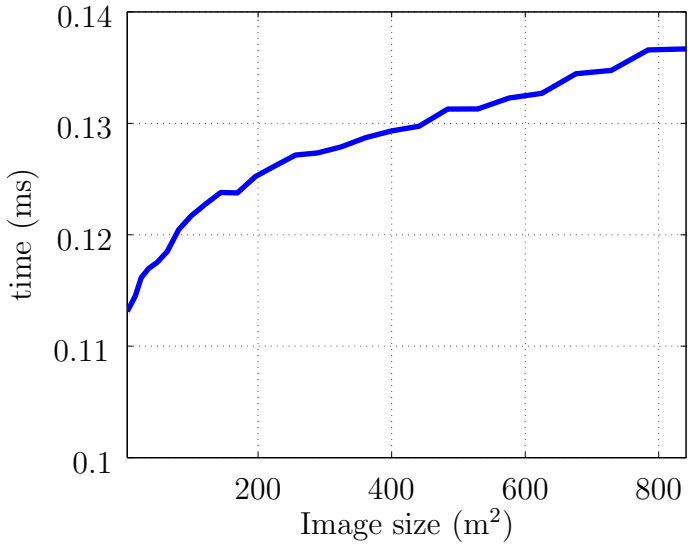


Figure 7.20: Processing time per pixel and per iteration for the ICM algorithm.

The size of the image influences the computational cost as well. For the ICM algorithm, when the image size increases by a certain factor, the processing time per pixel raises by a greater factor. This is represented in Fig. 7.20, where the time required by the algorithm per pixel and per iteration is depicted as a function of the image size. Hence, it is more efficient to segment relatively small sub-images of the original SAS images and then merge the results together, as already described in the previous section.

Chapter 8

Feature Extraction

ADAC systems follow the chain: segmentation, feature extraction, classification (see Fig. 2.1). Therefore, once the SAS images are segmented (see Chapter 7) and a database of S observations, i.e., detected objects, is available, each observation s, $1 \leq s \leq S$, is to be represented by a vector $\mathbf{t} = \{t_1, t_2, \ldots, t_N\}$. Each element t_j, $1 \leq j \leq N$, is referred to as feature, attribute or descriptor of the observation s. Subsequently, each object will be classified according to the comparison of its feature vector with the feature vectors of the observations in the training database (see Chapter 9).

Intuitively, a feature is good when it adopts very different values for observations belonging to different classes. However, it is the combination of several features that typically provides the best performance. In this chapter, a collection of features is proposed, which constitutes $\mathbf{t}$. The choice of the best subset $\mathbf{t}^* = \{t_1^*, t_2^*, \ldots, t_{n^*}^*\}$, $n^* < N$, accomplished by the feature selection algorithms described in Chapter 4, is regarded in Chapter 9.

The characterization of underwater objects can be done according to two main kinds of descriptors. On the one hand, the statistical properties of the sonar images can be exploited. On the other hand, the shape of the segmented regions are employed. Traditional sidescan ADAC systems are based on descriptors of the shadow rather than the highlight. This is due to the intensity variability that highlights present in sidescan images, which is less remarkable in SAS imagery [75]. In this thesis, highlight descriptors are also regarded. As described in Sec. 7.5, each object observation s in the database consists of the segmented shadow region and, if there is a highlight to the left of the shadow, also of the highlight region. For those observations lacking a highlight, the value of the highlight features are assigned to the sample mean of the training objects having a highlight, which assures that the missing features do not influence the classification of the observations.

Two databases of SAS images, SAS1 and SAS2, are considered in this thesis. Two kinds of man made objects exist in SAS1, spheres and cylinders. Ideally, the shadow of spherical objects exhibits a characteristic and invariant shape. It is always elongated and parallel to the range direction (see Fig. 7.17). Therefore, the descriptors of the shadow are expected to be useful in discerning spherical mines. On the other hand, the highlight of cylindrical objects is prominent, while the shape of the corresponding

shadow varies with the object orientation (see Fig. 7.16). Hence, highlight attributes are valuable for the characterization of cylinders. Truncated cones and wedge-shaped objects (in the SAS2 database) present more complex shapes than spheres and cylinders. They are better described by their shadow attributes.

Most of the works available in the literature perform the classification of the objects based on attributes of the shadow [90–92]. Some approaches focus on a single type of features, e.g., [13] employs Fourier coefficients, and [131] considers principal components. Other methods employ a very small number of attributes [14, 85].

In this chapter, statistical and geometrical descriptors are proposed. The latter characterize both the shadow and highlight shapes. Some sets of well-established shape descriptors sets are regarded: normalized central moments, invariant moments, Fourier coefficients and principal components. While some of them have been applied in sidescan applications, to the best knowledge of the author, this is the first time that they are tested on SAS images. Moreover, a group of heterogeneous features has been designed. They minimize the influence that have on them not only the object orientation with respect to the sonar system but also poor segmentation scenarios, which might be due to either low image quality or challenging seabeds such as sand ripples. The shadow descriptors take distinct values for the spheres, and the highlight features describe generally better the highlights of cylindrical objects. A novel feature that, based on a comparison of the segmentation results provided by the different methods (see Chapter 7), measures the segmentation result confidence, is particularly useful for discerning clutter objects.

This chapter is divided as follows. In Sec. 8.1 a set of statistical properties of the image is proposed as descriptors for the detected objects. Secs. 8.2 and 8.3 are devoted to sets of descriptors for both the shadow and highlight shapes. The feature comparing segmentation results is presented in Sec. 8.4. Secs. 8.5 to 8.8 refer to normalized central moments, invariant moments, principal components and Fourier coefficients, respectively. The estimated pdfs of some features for the SAS1 database are included in the Appendix. In Sec. 8.9, the feature extraction computational cost is regarded.

8.1 Statistical Features

Classification of underwater man made objects based on statistical features has been considered in the literature in several occasions. The main idea behind it is that the pixel intensity follows different distributions depending on the object class. Thus, the

intensity of shadow pixels is usually lower for man made objects than for clutter objects (see SAS images in Figs. 7.16 to 7.18). Moreover, highlight pixels tend to be lighter for man made than for clutter objects.

In [68], the difference of SNR between the different regions is employed for detection purposes. While the mean and variance are considered in [93], the kurtosis and skewness are used for detection and classification in [94,95]. In this thesis, a novel set of statistical features, which is based on a Weibull parametric model of the SAS images, is proposed.

As referred in Sec. 7.1.2.1, the different regions of an SAS image, `sdw`, `hlt` and `bkg`, are conveniently modeled by the Weibull distributions $\mathcal{W}(\xi_{\texttt{sdw}}, \xi'_{\texttt{sdw}})$, $\mathcal{W}(\xi_{\texttt{hlt}}, \xi'_{\texttt{hlt}})$ and $\mathcal{W}(\xi_{\texttt{bkg}}, \xi'_{\texttt{bkg}})$, respectively. The shadow and highlight parameters are estimated from the segmented regions. However, the background region might significantly vary from part to part of the SAS image, whose size is typically several orders of magnitude bigger than the objects of interest. Therefore, instead of using the entire background region to estimate the background Weibull parameters, only a stripe of pixels around the shadow of interest is used. The dilation morphological operation [103] is applied to the segmented shadow in order to build this stripe. See Fig. 8.1 for an illustration.

An alternative consists of using the difference between the parameters as features, that is, $\{\Delta\xi_{\texttt{bkg,sdw}}, \Delta\xi_{\texttt{sdw,hlt}}, \Delta\xi_{\texttt{hlt,bkg}}, \Delta\xi'_{\texttt{bkg,sdw}}, \Delta\xi'_{\texttt{sdw,hlt}}, \Delta\xi'_{\texttt{hlt,bkg}}\}$, where $\Delta\xi_{\texttt{bkg,sdw}} = \xi_{\texttt{bkg}} - \xi_{\texttt{sdw}}$, and equivalently for the other elements. In Fig. 8.2 the histogram of both $\xi_{\texttt{sdw}}$ and $\Delta\xi_{\texttt{bkg,sdw}}$ for the different classes of the SAS1 database, are depicted. Note that the value of $\Delta\xi_{\texttt{bkg,sdw}}$ is more class dependent and therefore, $\Delta\xi_{\texttt{bkg,sdw}}$ is presumably a better feature than $\xi_{\texttt{sdw}}$. Which features are the actual optimal ones for the databases at hand is determined by the feature selection algorithms (see Chapter 9). The histogram of the other statistical features are included in the Appendix, in Fig. A.1.

8.2 Shadow Geometrical Features

In this section, a set of heterogeneous descriptors for the shadow shape is presented. Some of them, e.g., the area, are standard and commonly used for representations of shapes. Others have been developed in this thesis for characterization of shadows in the context of mine hunting, taking into account that invariance to changes of object orientation with respect to the sonar system and resistance to poor segmentation scenarios are desirable characteristics for the descriptors. The distribution of all features presented in this section for the SAS1 database are depicted in Fig. A.2.

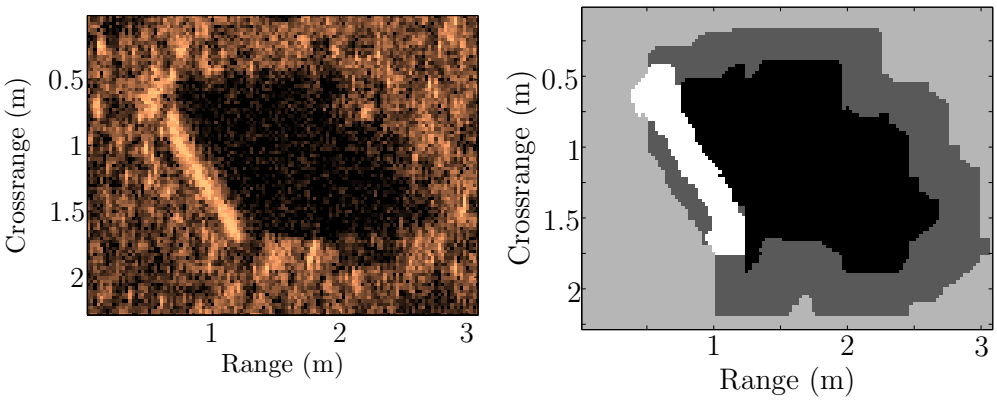


Figure 8.1: Snapshot of SAS image showing a cylindrical object (left) and estimated label field (right), where a stripe around the shadow indicates the pixels used for the background statistics estimation.

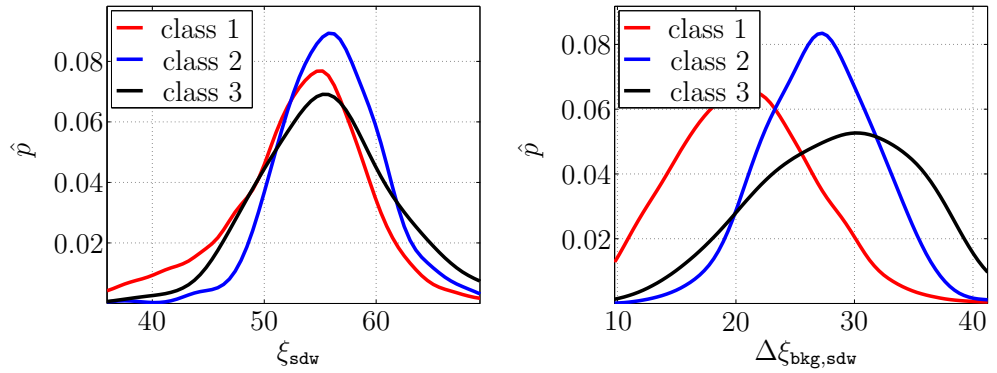


Figure 8.2: Estimated distribution of statistical features for the three classes in the SAS1 database. The clutter objects correspond to class 1 (red), while the spherical and cylindrical man made objects are assigned to classes 2 (blue) and 3 (black), respectively.

The literature offers a large number of descriptors for shape representation [103, 132]. They can be divided into two main groups: those that model the contour of the region of interest, and those that model the region itself. For sonar images, the latter are more appropriate, since the variability of the contour is not meaningful.

Two straightforward features are the area $\gamma_{\mathtt{sdw}}$ and perimeter ρ of the shadow. The ratio between the perimeter squared and the area, known as compactness $\chi = \frac{\rho^2}{\gamma_{\mathtt{sdw}}}$ [133], is also considered. It has been employed for mine hunting applications in [90, 91]. The compactness is minimum for a circle and tends to infinity as the shape approaches a straight line. Hence, it reaches high values for the elongated shadow of the sphere class.

Two other features, the ratio of principal axes $r_{\mathtt{sdw}}$ and the orientation o, also have distinct values for spherical objects. The principal axes of a region are defined as

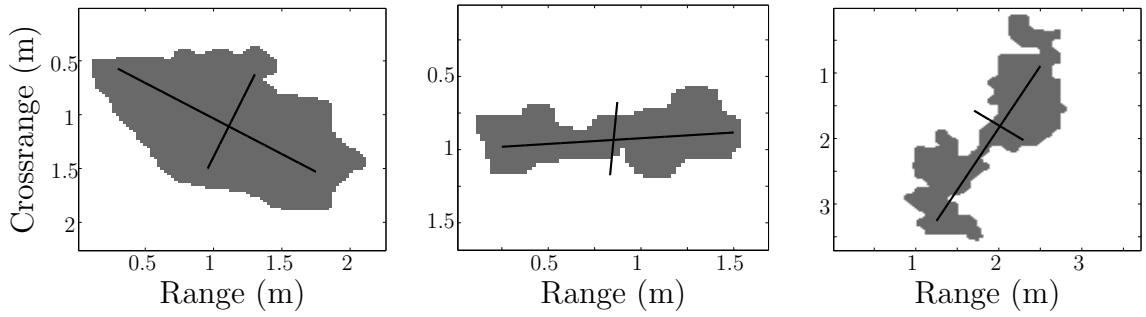


Figure 8.3: Principal axes of the shadow of a cylindrical, spherical and clutter object, respectively. While the principal axes have more or less random position for both cylindrical and clutter objects, they tend to be parallel to the Cartesian axes for the shadows of spherical objects.

two line segments that cross orthogonally in the center of mass and represent the directions with zero cross-correlation (see Fig. 8.3). For a given region with contour $\mathbf{b} = \{\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_V\}^{\mathrm{T}}$ and center of mass $\mathbf{c_b}$, the covariance matrix of the contour is defined by

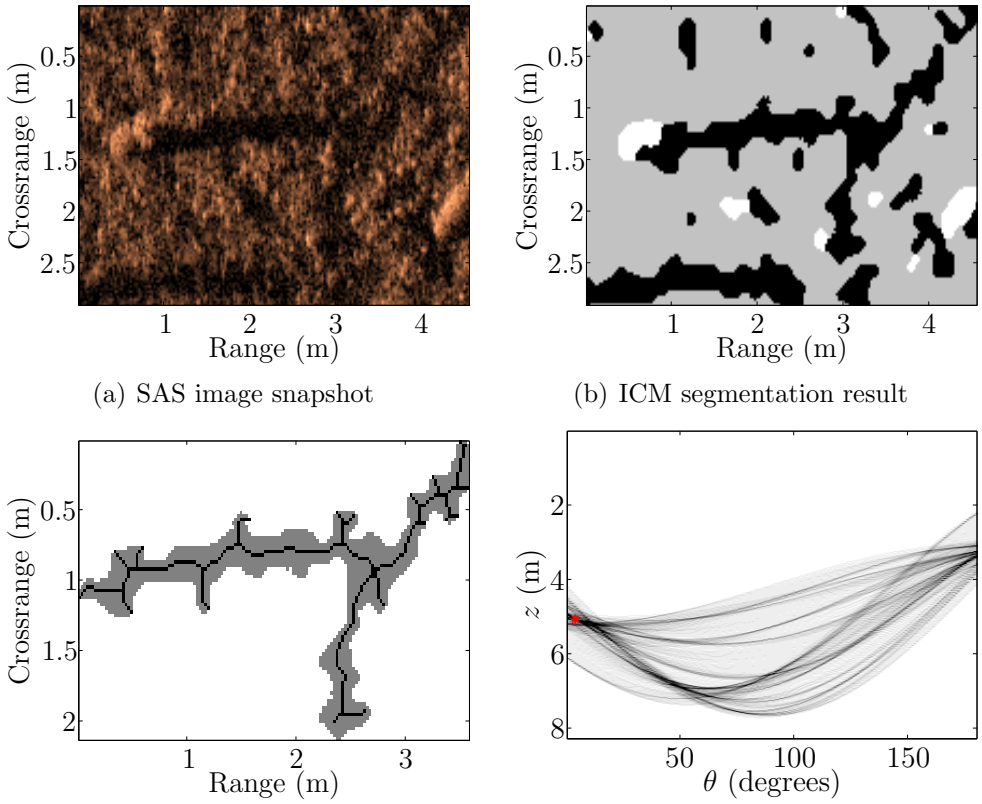
$$\mathbf{\Sigma} = \sum_{j=1}^{V} (\mathbf{b}_j - \mathbf{c_b})(\mathbf{b}_j - \mathbf{c_b})^{\mathrm{T}}. \tag{8.1}$$

The nodes $\mathbf{b}_j$ and the center of mass $\mathbf{c_b}$ are expressed in Cartesian coordinates. The ratio of principal axes, $r_{\mathtt{sdw}}$, equals the coefficient of the eigenvalues of $\mathbf{\Sigma}$ and is a good measure of the elongation of the region defined by $\mathbf{b}$.

Normally, the orientation o is measured as the angle that the major axis of the shadow forms with the abscissa of the image. However, in order to increase the robustness of the orientation estimation with respect to challenging seabeds scenarios, the following approach has been adopted. Given the binary representation of the segmented shadow, $\mathbf{I}$, its topological skeleton Λ [103] is extracted and the Hough transform [134] of Λ, Υ, is calculated. Loosely speaking, the skeleton of a shape corresponds to a thin version of this shape that is equidistant to its contour. Each point of Λ with Cartesian coordinates $\{u, v\}$ contributes to Υ at a certain z (distance to the origin of coordinates) and θ (angle with respect to the abscissa), according to $z = u \cdot \cos\theta + v \cdot \sin\theta$. The range direction corresponds to u, and v is parallel to the crossrange direction. We assign

$$o = \arg\max_{\theta}\{\Upsilon(z, \theta)\}, \tag{8.2}$$

which estimates the orientation of Λ, and therewith of the shadow region. An example is shown in Fig. 8.4. The shadow of the sand ripples are segmented together with the shadow of the spherical mine. This is, indeed, a common scenario that represents one of the greatest challenges for ADAC of underwater objects [73]. The proposed skeleton based approach allows for a correct measure of the shadow orientation. Fig. 8.5 shows



(a) SAS image snapshot

(b) ICM segmentation result

(c) Binary representation of the segmented shadow, **I**, and its skeleton

(d) Hough transform of the skeleton

Figure 8.4: Measure of the orientation o of the shadow of a spherical man made object based on the Hough transform of the shadow topological skeleton. The red dot in the Hough transform diagram (Fig. 8.4(d)) indicated the position of the maximum, i.e., $o = 4$ degrees. Note that a local maximum appears at $\theta = 48$. This corresponds to the orientation of the sand ripple shadow segmented together with the sphere shadow. As long as the sphere shadow is longer than the sand ripple, the absolute maximum of Υ will correspond to the orientation of the sphere shadow and therefore, it will be correctly estimated.

the distribution of the orientation as estimated by the angle of the shadow major axis (left) and as estimated by the proposed skeleton based method (right). The histograms of clutter and cylindrical objects do not significantly vary. However, the distribution of the orientation for spherical objects shows a higher concentration around 0 degrees for the skeleton based method, which is in agreement with the fact that shadows of spheres are parallel to the range direction.

The width of the shadow region in both range and crossrange directions, η_u and η_v, respectively, have also been employed. Note that, if directly measured from the segmented shadow, the value of η_v is very sensitive to poor segmentation scenarios. For example, the crossrange width of the segmented shadow **I** in Fig. 8.4(c) is $\eta_v > 2$ meters. However, the 'real' width is not greater than 0.7 meters. The following approach

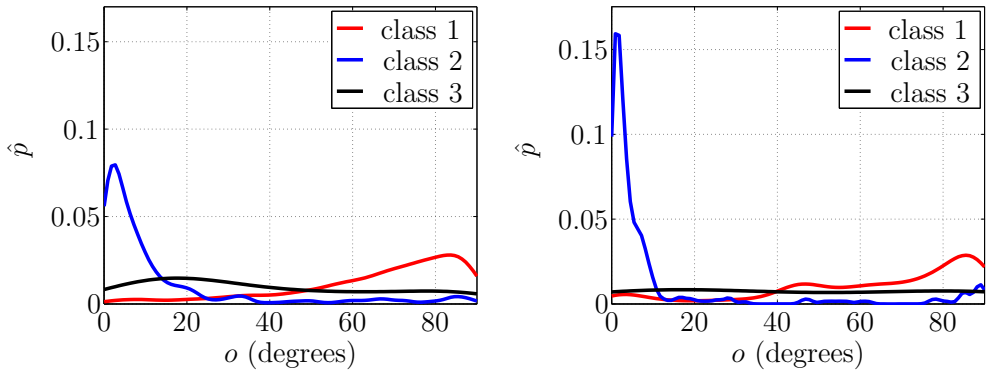


Figure 8.5: Estimated distribution of the shadow orientation o when computed directly from the segmented shadow (left) and when measured from the shadow skeleton (right). The clutter objects correspond to class 1 (red), while the spherical and cylindrical man made objects are assigned to classes 2 (blue) and 3 (black), respectively.

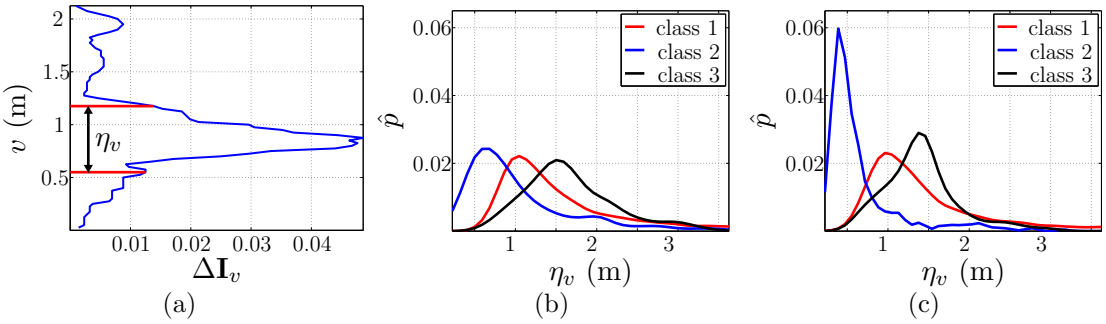


Figure 8.6: Crossrange width, η_v. Fig. 8.6(a) illustrates the measure of η_v for the segmented shadow in Fig. 8.4. It is based on the cumulative function $\Delta\mathbf{I}_v$. The values v_1 and v_1 are indicated by two red lines. Figs. 8.6(b) and 8.6(c) depict the pdfs of η_v for the different classes in SAS1 if η_v is estimated directly from the segmented shadow and if the measure is based on $\Delta\mathbf{I}_v$, respectively.

is proposed in order to improve the measurement. The function

$$\Delta\mathbf{I}_v = \frac{1}{\bar{\Delta}} \sum_u I_{uv}, \tag{8.3}$$

is defined, where I_{uv} is the element of $\mathbf{I}$ with coordinates u and v and $\bar{\Delta} = \sum_u \sum_v I_{uv}$ is a normalization constant. Basically, $\Delta\mathbf{I}_v$ consists on the cumulative projection of $\mathbf{I}$ on the crossrange axis. We define

$$v_1 = \min\{v\} \quad | \quad \Delta\mathbf{I}_v > \frac{1}{4} \cdot \max\{\Delta\mathbf{I}_v\} \tag{8.4}$$

$$v_2 = \max\{v\} \quad | \quad \Delta\mathbf{I}_v > \frac{1}{4} \cdot \max\{\Delta\mathbf{I}_v\}, \tag{8.5}$$

and finally, the more accurate estimation of the crossrange width is calculated as $\eta_v = v_2 - v_1$. For the example in Fig. 8.4(c), $\eta_v = 0.62$ meters. Fig. 8.6(a) illustrates $\Delta\mathbf{I}_v$,

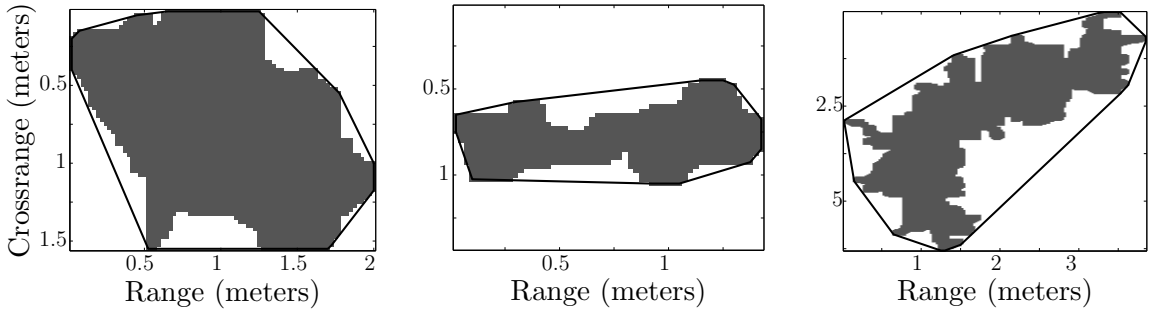


Figure 8.7: Solidity of the shadow of a cylindrical, spherical and clutter objects, respectively. Each shadow region is surrounded by the contour of its minimal convex hull.

v_1 and v_2. For man made objects that have been correctly segmented, the proposed technique is practically equivalent to a direct measurement of η_v. The distribution of η_v for the different classes in the SAS1 data set, for both a direct measurement and a measurement based on the cumulative function $\Delta\mathbf{I}_v$, are depicted in Figs. 8.6(b) and 8.6(c), respectively. It can be observed that the crossrange width for the sphere class (class 2) is significantly narrower when the proposed measure is employed.

The ratio between both widths as a feature, $r_\eta = \eta_v/\eta_u$ and the amount $\Psi = \max\{\Delta\mathbf{I}_v\}/\eta_v$, have proven as valuable descriptors as well.

The solidity Γ is the coefficient between the area of the region and the minimal convex area that comprises it [133]. Many shadows of clutter objects have a very low solidity. The solidity of the clutter shadow in Fig. 8.7 equals 0.61. It is 0.85 for the cylinder and 0.72 for the sphere.

Table 8.1 summarizes the ten shadow geometrical descriptors.

8.3 Highlight Geometrical Features

In this section, some simple geometrical features of the highlight are included. Furthermore, a set of features that describe the relation between shadow and highlight are described. Their distributions for the SAS1 database are included in Fig. A.3.

While almost 35 % of the clutter objects lack a highlight, more than 90 % of the man made objects, for both data sets, have a highlight associated to the segmented shadow. Therefore, the presence, $\epsilon_{\texttt{hlt}} = 1$, or absence, $\epsilon_{\texttt{hlt}} = 0$, of a highlight is already a valuable feature.

Feature	Description
$\gamma_{\texttt{sdw}}$	area
ρ	perimeter
χ	compactness
$r_{\texttt{sdw}}$	ratio of principal axes
o	orientation
η_u	range width
η_v	crossrange width
r_η	ratio between η_u and η_v
Ψ	$\max\{\Delta\mathbf{I}_v\}/\eta_v$
Γ	solidity

Table 8.1: Shadow geometrical features

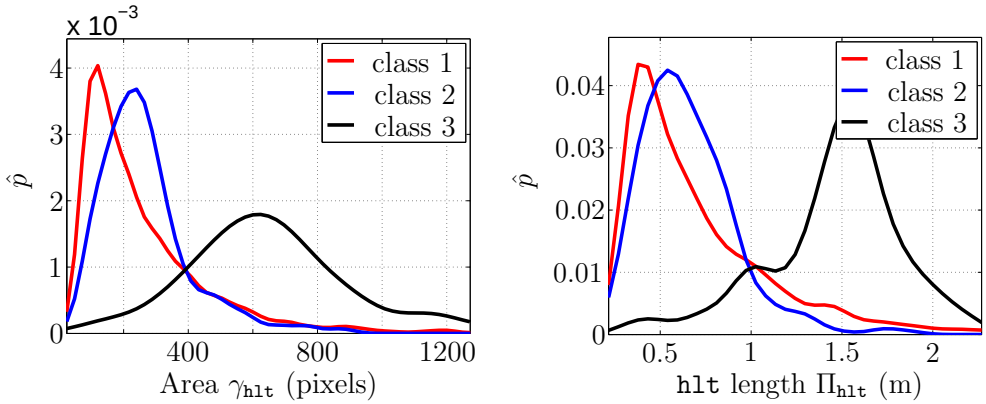


Figure 8.8: Distribution of geometrical highlight features. The clutter objects correspond to class 1 (red), while the spherical and cylindrical man made objects are assigned to classes 2 (blue) and 3 (black), respectively.

The area of the highlight $\gamma_{\texttt{hlt}}$ and the crossrange length $\Pi_{\texttt{hlt}}$ are simple but meaningful descriptors. As depicted in Fig. 8.8, both reach significantly high values for the cylindrical man made objects (class 3). Another meaningful descriptor for discerning cylindrical objects is the ratio of principal axes of the highlight, $r_{\texttt{hlt}}$.

Three descriptors characterize the highlight-shadow relation. First, the rate between the highlight and shadow widths along the crossrange direction, $r_{\texttt{sdw,hlt}}$, is considered. It is generally close to one for mines, while it might significantly differ from this value for clutter objects. The mean distance between highlight and shadow, $d_{\texttt{sdw,hlt}}$, and the difference between the orientation of the right part of the highlight contour and the orientation of the left part of the shadow contour, Δ_o, have been regarded (see Fig. 8.9). The influence of the highlight orientation on the orientation of the shadow

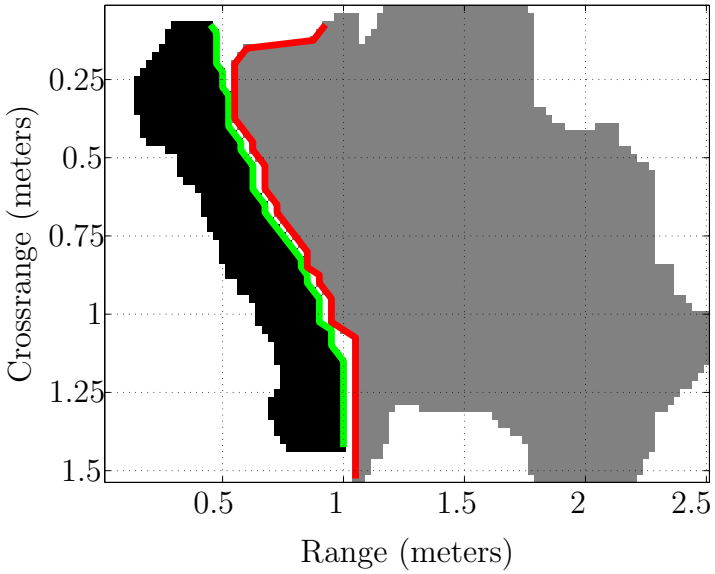


Figure 8.9: Difference between the orientation of the right part of the highlight contour (in green) and the orientation of the left part of the shadow contour (in red) of a cylinder. For cylindrical objects, this feature takes smaller values.

Feature	Description
$\epsilon_{\texttt{hlt}}$	presence of highlight
$\gamma_{\texttt{hlt}}$	area
$\Pi_{\texttt{hlt}}$	crossrange length
$r_{\texttt{hlt}}$	ratio of principal axes
$r_{\texttt{sdw,hlt}}$	rate between highlight and shadow widths
$d_{\texttt{sdw,hlt}}$	mean distance between shadow and highlight
Δ_o	shadow and highlight orientation distance

Table 8.2: Highlight geometrical features and highlight-shadow relation features.

is stronger for cylindrical objects.

For those objects with no associated highlight, all highlight features are assumed to equal the average feature value of the training data set.

Table 8.2 summarizes the seven features that describe the highlight and the highlight-shadow relation.

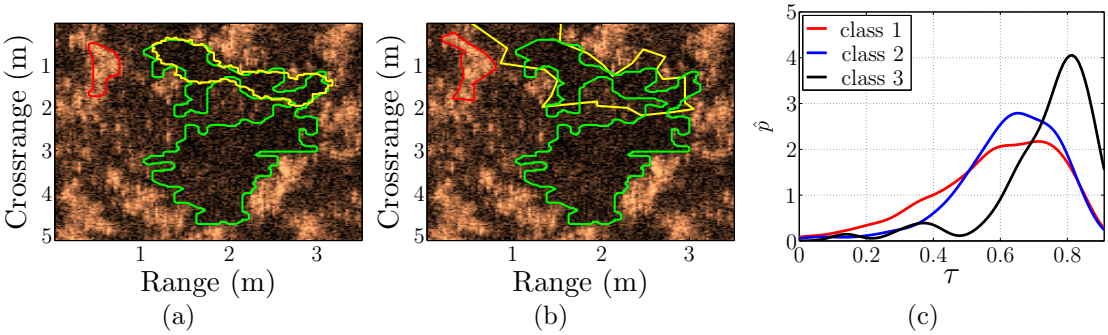


Figure 8.10: Segmentation overlap. The segmentation results provided by the ICM (green) and min-cut/max-flow (yellow) algorithms for a clutter object are shown in Fig. 8.10(a). Fig. 8.10(b) superimposes the ICM (green) and AC (yellow) segmentation results. For shadow regions that, like this one, are not well differentiated from the background, the algorithms produce significantly different segmentations. Fig. 8.10(c) shows the distribution of τ for the SAS1 database.

8.4 Segmentation Overlap

When the shadow and highlight of an object are prominent with respect to the background, all three segmentation algorithms, ICM, AC and min-cut/max-flow, provided similar segmentation results (see, for example, the three first rows of Fig. 7.16). On the contrary, if the difference among regions is diffuse, the algorithms tend to provide significantly distinct results. This is prone to happen when the irregularities of the seabed prompt the appearance of clutter (see last two rows of Fig. 7.18).

The AC algorithm is initialized after the ICM segmentation result (see Sec. 7.4.2). This means that, when the AC segmentation is computed, the ICM segmentation result is available as well. The ratio τ between the area of the region where both segmentation results intersect and the area of the AC segmented region constitutes a good measure of the segmentation reliability.

This feature can be analogously calculated for the min-cut/max-flow algorithm, whose initialization is also based on the ICM result (see Sec. 7.3.3). Fig. 8.10 shows the ICM segmentation of a clutter shadow and, superimposed, the min-cut/max-flow (Fig. 8.10(a)) and the AC (Fig. 8.10(b)) results. They significantly differ. In Fig. 8.10(c) the distribution of τ for the SAS1 data set is depicted. Cylindrical objects are segmented similarly by all algorithms and hence $\tau \lesssim 1$ for class 3 (cylinders). As expected, the value of τ is lower for class 1 objects (clutter). The segmentation results of spherical objects are also significantly different for the ICM algorithm and the AC or min-cut/max-flow algorithms (see Fig. 7.17), which is corroborated by τ.

8.5 Normalized Central Moments

Normalized central moments were introduced as features for underwater objects classification in [91]. The normalized central moments of the binary representation of the segmented shadow are considered as features in this thesis.

For an image $\mathbf{I}$, the sample central moment of order $(i+j)$ is defined as

$$\hat{\sigma}_{ij} = \sum_u \sum_v (u - c_u)^i (v - c_v)^j I_{uv}. \tag{8.6}$$

The origin of coordinates is placed at the center of mass of the image, $\mathbf{c_I} = \{c_u, c_v\}$. Due to this normalization, $\hat{\sigma}_{10}$ and $\hat{\sigma}_{01}$ vanish. For a binary image that takes its values from $\{0, 1\}$, $\hat{\sigma}_{00}$ equals the area $\gamma_{\texttt{sdw}}$ of the object in $\mathbf{I}$.

The normalized central moments are obtained normalizing $\hat{\sigma}_{ij}$ with respect to the area

$$\bar{\sigma}_{ij} = \frac{\hat{\sigma}_{ij}}{\hat{\sigma}_{00}^{1+\frac{i+j}{2}}}, \tag{8.7}$$

The moments of order 2 have a simple geometrical interpretation [135]. While $\bar{\sigma}_{11}$ is related with the covariance of the region, $\bar{\sigma}_{20}$ and $\bar{\sigma}_{02}$ correspond respectively to the length of the major and minor axes of the ellipse that best fits the object in $\mathbf{I}$.

The normalized central moments $\bar{\sigma}_{ij}$ of the segmented shadow up to order 10 have been regarded. Their distributions for the SAS1 data set are included in Fig. A.4.

8.6 Invariant Moments

The normalized central moments are invariant to scale and translation, but not to rotation. The following moments, known as invariant or Hu moments [136] are invariant

to all scale, translation and also rotation:

$$\iota_1 = \bar{\sigma}_{20} + \bar{\sigma}_{02} \tag{8.8}$$

$$\iota_2 = (\bar{\sigma}_{20} - \bar{\sigma}_{02})^2 + (2\bar{\sigma}_{11})^2 \tag{8.9}$$

$$\iota_3 = (\bar{\sigma}_{30} - 3\bar{\sigma}_{12})^2 + (3\bar{\sigma}_{21} - \bar{\sigma}_{03})^2 \tag{8.10}$$

$$\iota_4 = (\bar{\sigma}_{30} + \bar{\sigma}_{12})^2 + (\bar{\sigma}_{21} + \bar{\sigma}_{03})^2 \tag{8.11}$$

$$\begin{aligned}\iota_5 = {} & (\bar{\sigma}_{30} - 3\bar{\sigma}_{12})(\bar{\sigma}_{30} + \bar{\sigma}_{12})[(\bar{\sigma}_{30} + \bar{\sigma}_{12})^2 - 3(\bar{\sigma}_{21} + \bar{\sigma}_{03})^2] + \\ & (3\bar{\sigma}_{21} - \bar{\sigma}_{03})(\bar{\sigma}_{21} + \bar{\sigma}_{03})[3(\bar{\sigma}_{30} + \bar{\sigma}_{12})^2 - (\bar{\sigma}_{21} + \bar{\sigma}_{03})^2]\end{aligned} \tag{8.12}$$

$$\iota_6 = (\bar{\sigma}_{20} - \bar{\sigma}_{02})[(\bar{\sigma}_{30} + \bar{\sigma}_{12})^2 - (\bar{\sigma}_{21} + \bar{\sigma}_{03})^2] + 4\bar{\sigma}_{11}(\bar{\sigma}_{30} + \bar{\sigma}_{12})(\bar{\sigma}_{21} + \bar{\sigma}_{03}) \tag{8.13}$$

$$\begin{aligned}\iota_7 = {} & (3\bar{\sigma}_{21} - \bar{\sigma}_{03})(\bar{\sigma}_{30} + \bar{\sigma}_{12})[(\bar{\sigma}_{30} + \bar{\sigma}_{12})^2 - 3(\bar{\sigma}_{21} + \bar{\sigma}_{03})^2] - \\ & (\bar{\sigma}_{30} - 3\bar{\sigma}_{12})(\bar{\sigma}_{21} + \bar{\sigma}_{03})[3(\bar{\sigma}_{30} + \bar{\sigma}_{12})^2 - (\bar{\sigma}_{21} + \bar{\sigma}_{03})^2]\end{aligned} \tag{8.14}$$

All seven invariant moments have been included in the feature vector $\mathbf{t}$. Their distributions for the SAS1 database are included in Fig. A.5.

8.7 Principal Components Analysis

Principal Components Analysis (PCA) [137] is a popular pattern recognition technique. It has been studied for sidescan sonar detection in [131], where PCA is applied directly to the sonar images. By contrast, in this thesis it is used on the binary representation of the shadow region, $\mathbf{I}$.

PCA is a tool to represent a set of correlated variables by a smaller number of uncorrelated ones, called principal components. The first principal component, ζ_1, represents as much of the data correlation as possible, and each consecutive ζ_j accounts for as much of the remaining correlation as possible. For implementation details see [103,138].

In this application, the correlated variables are the pixels of the binary representation of the segmented shadow, normalized in size around its center of mass. A total of 50 PCA coefficients, $\{\zeta_j\}$, $1 \leq j \leq 50$, have been considered. The distributions of the first twelve ζ_j for the SAS1 data set are included in Fig. A.6.

8.8 2D-Fourier Descriptors

The utilization of Fourier coefficients as descriptors of underwater objects has been applied in [13, 139]. They are also considered in this thesis. The 2D-Fourier transform

is applied to the binary representation of the segmented shadow, $\mathbf{I}$, normalized in size around the center of mass. The Fourier coefficients $\{\Xi_{i,j}\}$, $0 \leq i \leq 7$, $0 \leq j \leq 7$, are included in $\mathbf{t}$. Higher order coefficients are not necessary, since they correspond to the high frequency components in the image, which are related to noise rather than to significant shape information, concentrated in lower frequencies. The distributions of some $\Xi_{i,j}$ for the SAS1 data set are depicted in Fig. A.7.

8.9 Computational Cost

The cost of computing the feature set $\mathbf{t}$ is considered in this section. Some features require the same amount of time independently of the object size, e.g., the 2D-Fourier coefficients, which are calculated on the normalized shadow. By contrast, other features need more time for bigger object sizes. Hence, it is logical that the spherical objects, which are smaller than the cylindrical objects (both highlight and shadow), have a smaller computational cost. For the SAS1 database, the average computational cost for spherical, cylindrical and clutter objects is 0.15, 0.23 and 0.18 seconds, respectively. An Intel i5 4 core 2.8 GHz processor has been employed for the simulations.

Chapter 9

Classification and Feature Selection

The design procedure for mine hunting ADAC systems proposed in this thesis is summarized in Fig. 9.1. First, the SAS image is segmented (see Chapter 7). Each object constitutes an observation s of the database, $1 \leq s \leq S$, which is characterized by a feature vector $\mathbf{t}$ (see Chapter 8). After selecting a set of classifier candidates (e.g., Mahalanobis' classifier and k-Nearest Neighbor), the resampling method (proposed in Chapter 3) assesses their performance and the best classifier is chosen. Moreover, confidence intervals for the optimal number of features n^* are provided. Finally, the optimal feature subset $\mathbf{t}^*$ is found by the feature selection algorithms, the D-SFS or the D-SFFS (presented in Chapter 4). This chapter provides the results obtained for the two SAS databases at hand, SAS1 and SAS2, by the last two blocks, namely, the resampling method and the feature selection algorithm.

The SAS1 data set consists of 437 man made objects (308 spheres and 129 cylinders) and 3604 clutter objects. The SAS2 database comprises 45 wedge-shaped objects, 67 cylinders and 73 truncated cones. The snapshots are small and centered on the objects and, therefore, almost no clutter was obtained. In order to study the ability of the system to avoid false alarms though, the 3604 clutter elements from the first data set have been adopted by the second database as well. Note that the imbalance between the clutter class and the mine classes is very strong. Hence, employing the overall misclassification probability as figure of merit for the ADAC system focuses on the minimization of the false alarm rate. Indeed, the focus of a mine hunting system should be on the minimization of the missed detected mines (mines classified as clutter). This issue is regarded in the definition of a suitable figure of merit for the system.

Both databases are multiclass, that is, they consists of more than two classes. In such cases, it is sometimes advantageous to utilize a cascade configuration of binary classification systems. Because the false alarm rate provided by the standard multiclass configuration for the SAS1 database is not outstanding, such scheme has been investigated. A first classifier distinguishes the spherical mines and, subsequently, a second one divides the remaining objects into two classes, cylinders and clutter.

All classification results referred above stem from the min-cut/max-flow segmentation. The quality assessment for classifier performance has been employed to compare them with the results provided by the other segmentation algorithms presented in Chapter

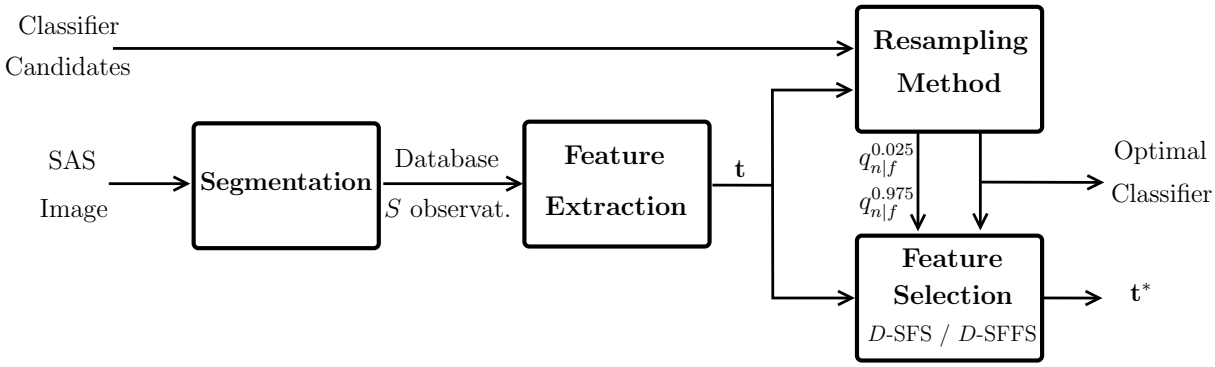


Figure 9.1: Scheme of the design procedure of the ADAC system for mine hunting proposed in this thesis. The outputs of the system, $\mathbf{t}^*$ and the optimal classifier, will be employed by the working ADAC system consisting of the segmentation, feature extraction and classification steps (see Fig. 2.1).

7, the ICM and AC algorithms. The influence of both the classification system and the segmentation method on the classification results are compared.

Finally, the computational cost of both the resampling method and the feature selection algorithms for the SAS databases is investigated.

This chapter is organized as follows. Sec. 9.1 proposes a figure of merit alternative to the overall misclassification probability. The quality of four classification systems, k-NN, Mahalanobis' classifier, LDA and SVM for the SAS1 and SAS2 databases has been assessed and results are shown in Sec. 9.2. Further, confidence intervals for the optimal number of features are obtained. The performance of the D-SFS and D-SFFS feature selection algorithms is illustrated in Sec. 9.3. Besides the straightforward three class classification scheme, a two class cascade configuration has been considered for the SAS1 data set. In Sec. 9.4, the classification results provided by the three segmentation algorithms regarded in this thesis are compared. The computational cost of the design procedure is tackled in Sec. 9.5.

9.1 Figure of Merit

The natural figure of merit f of a classifier is the overall misclassification rate, P_{m} (see Sec. 3.4). Although P_{m} is bounded for some specific distributions, the misclassification rate must be estimated from the available data if the distribution is unknown and the number of observations is finite. Given a database of test observations and a class label

Table 9.1: Confusion matrix of a 3 class system. The missed detected mines are indicated in red and the false alarms in blue.

		Predicted		
		class 1	class 2	class 3
Actual	class 1	$z_{1\|1}$	$z_{2\|1}$	$z_{3\|1}$
	class 2	$z_{1\|2}$	$z_{2\|2}$	$z_{2\|2}$
	class 3	$z_{1\|3}$	$z_{2\|3}$	$z_{3\|3}$

$c \in \{1, \ldots, C\}$ associated with each observation, where C is the number of classes, the misclassification rate is calculated as

$$P_{\mathrm{m}} = \sum_{c_1} P(c_1) \cdot \sum_{c_2 \neq c_1} P(c_2|c_1), \quad c_1, c_2 \in \{1, \ldots, C\}, \tag{9.1}$$

where $P(c_1)$ is the prior probability of class c_1 and $P(c_2|c_1)$ is the probability of deciding for class c_2 when the actual class is c_1. The 5-fold cross validation approach (see Sec. 3.4) has been adopted in order to estimate f from the available observations.

Let us now consider our particular problem. The SAS databases consist of 3604 clutter objects (class 1) and either two (database SAS1) or three (database SAS2) other classes with a number of observations between 45 and 308. Hence, the problem is a multiclass one and imbalanced. Given the dominance of class 1, minimizing $f := P_{\mathrm{m}}$ focuses on reducing $P(c|1)$ with $c \neq 1$, that is, the false alarm rate. Indeed, we are far more interested in reducing $P(1|c)$, i. e., the rate of mines classified as clutter or missed detected mines. In the sequel, we propose a new figure of merit that solves this issue.

The confusion matrix of a system with $C = 3$ classes is represented in Table 9.1. Element $z_{c_2|c_1}$ accounts for the number of elements with actual class c_1 and predicted class c_2. The expression of the proposed figure of merit reads:

$$f_\lambda = \lambda \sum_{c_2 \neq 1} z_{c_2|1} + (1 - \lambda) \sum_{c_1 \neq 1} \sum_{c_2 \neq c_1} z_{c_2|c_1}, \tag{9.2}$$

for $c_1, c_2 \in \{1, \ldots, C\}$ and $0 \leq \lambda \leq 1$. There are two main differences between Eqs. (9.1) and (9.2). First, the former expresses all quantities as probabilities while the latter employs absolute numbers of observations. Secondly, the former weighs the classification error of each class with its prior probability while the latter introduces a weighting factor λ that determines the relative importance of misclassifying clutter with respect to misclassifying a mine, independently of the prior distributions of the classes.

Note that both $f := P_{\mathrm{m}}$ and $f := f_\lambda$ give the same importance to classifying a mine class observation $c_1 \neq 1$ as a different mine class, and to classifying it as clutter (missed detection). That is, in Eq. (9.1), the weight of $P(c_2|c_1)$ is the same for $c_2 \neq 1$ (mine) and $c_2 = 1$ (clutter), and analogously in Eq. (9.2). A new weighting factor λ' allows for distinguishing both scenarios:

$$\begin{aligned} f_{\lambda,\lambda'} &= \lambda \underbrace{\sum_{c_2 \neq 1} z_{c_2|1}}_{\text{false alarms}} \\ &+ (1-\lambda)\Big[\lambda' \underbrace{\sum_{c_1 \neq 1} \sum_{c_2 \neq 1, c_1} z_{c_2|c_1}}_{\text{mines wrong class}} + (1-\lambda') \underbrace{\sum_{c_1 \neq 1} z_{1|c_1}}_{\text{missed mines}} \Big], \end{aligned} \tag{9.3}$$

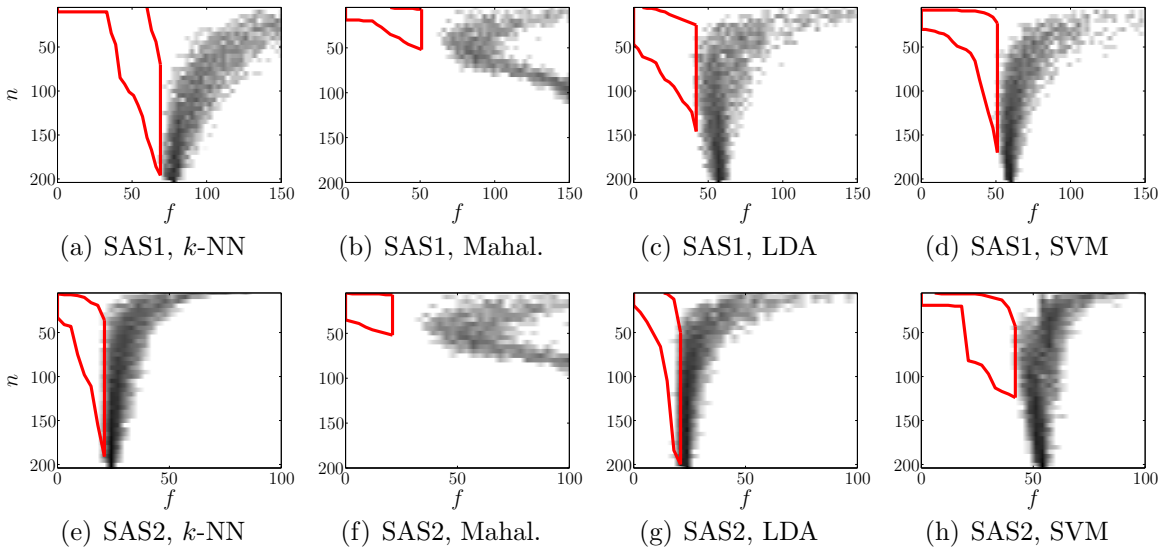
for $c_1, c_2 \in \{1, \ldots, C\}$ and $0 \leq \lambda, \lambda' \leq 1$. For $\lambda' = 0$, the assignment of a mine to a wrong class is not penalized as long as it is a mine class (i.e., it is not clutter).

9.2 Classifier Performance Assessment and Optimal Number of Features

The classifier performance assessment presented in Chapter 3 has been employed to compare the following classifier candidates for both data sets: a k-NN classifier with $k = 5$ [7], Mahalanobis' classifier [31], the Linear Discriminant Analysis classifier (LDA) [7] and Support Vector Machines (SVM) with a radial basis kernel [46].

As figure of merit, the expression proposed in Eq. (9.3), $f := f_{\lambda,\lambda'}$, has been adopted. The focus is on the minimization of the missed detected mines while keeping a reasonably low false alarm rate. Therefore, the relative weighting of the false alarms has been fixed to $\lambda = 0.1$ for both data sets. While $\lambda' = 0$ for the first database (no penalty is associated with assigning a wrong mine class to a mine observation), $\lambda' = 0.5$ for the SAS2 database (the same importance is given to missed detecting a mine as to classifying it as a mine of a wrong type).

Fig. 9.2 shows the estimated $\hat{p}_{f|n_i}$, $n_i \in \{1, \ldots, N\}$, for both data sets and all four classifier candidates. A logarithmic scale has been employed, which spans between -25 dB (white) and -5 dB (black) for all examples. Each horizontal line corresponds to $\hat{p}_{f|n_i}$ for a certain n_i, and has been obtained by histogram techniques from the figure



(a) SAS1, k-NN (b) SAS1, Mahal. (c) SAS1, LDA (d) SAS1, SVM

(e) SAS2, k-NN (f) SAS2, Mahal. (g) SAS2, LDA (h) SAS2, SVM

Figure 9.2: Quality assessment for classifier performance. The curves represent $10 \cdot \log(\hat{p}_{f|n_i})$ for $n_i = 1, \ldots, N$, estimated by histogram techniques from the figure of merit estimates. Both the database and the classification approach are indicated for each figure. The scale is common to all figures and spans between -25 dB (white) and -5 dB (black). A red line delimits the most probable region for the pair $\{f^*, n^*\}$, corresponding to $f \leq f_b^{\min}$ and $q_{n|f}^{0.025} \leq n \leq q_{n|f}^{0.975}$.

of merit estimates $\{f'_{1,n_i}, \ldots, f'_{N_B,n_i}\}$. Choosing a specific value of f, $\hat{p}_{n|f}$ is observed along each vertical line (only lacking the normalization constant A). The more to the left the distribution energy is, the better the classifier performs. The $f - n$ region delimited by the quantile $f \leq f_b^{\min}$ and the confidence interval $q_{n|f}^{0.025} \leq n \leq q_{n|f}^{0.975}$ is indicated by a red line.

The curse of dimensionality (see Sec. 3.1) is clearly visible for Mahalanobis' classifier for both data sets (see Figs. 9.2(b) and 9.2(f)), i.e., the figure of merit improves as n increases until a certain point and it degrades subsequently. Although the mean of $\hat{p}_{f|n_i}$, $n_i \in \{1, \ldots, N\}$, might decrease with n (e.g., Fig. 9.2(a)), which suggests that lower f values will be reached at higher n, the variance of $\hat{p}_{f|n_i}$ decreases as well with n. Therefore, the energy concentrated in the left tail of $\hat{p}_{f|n_i}$, where the optimal f^* will eventually be found, is smaller for values of n close to N. For this reason the confidence intervals indicate that n^* will most probably be placed in the lower n regions.

The values of the quality assessment Q with $\psi = 1$ (see Eq. (3.20)) for all classifiers and both data sets are included in Table 9.2. The results are in agreement with the curves in Fig. 9.2, that is, higher Q values correspond to distributions whose energy is concentrated in smaller values of f. Mahalanobis' classifier shows the worse performance for both examples. On the other hand, the LDA is the best candidate for

Table 9.2: Performance assessment Q, with $\psi = 1$, of the k-NN classifier with $k = 5$, Mahalanobis' classifier, LDA and SVM approach. For each data set the best result has been highlighted.

	SAS1	**SAS2**
k-NN	0.010	0.035
Mahal.	0.007	0.024
LDA	**0.016**	**0.037**
SVM	0.014	0.019

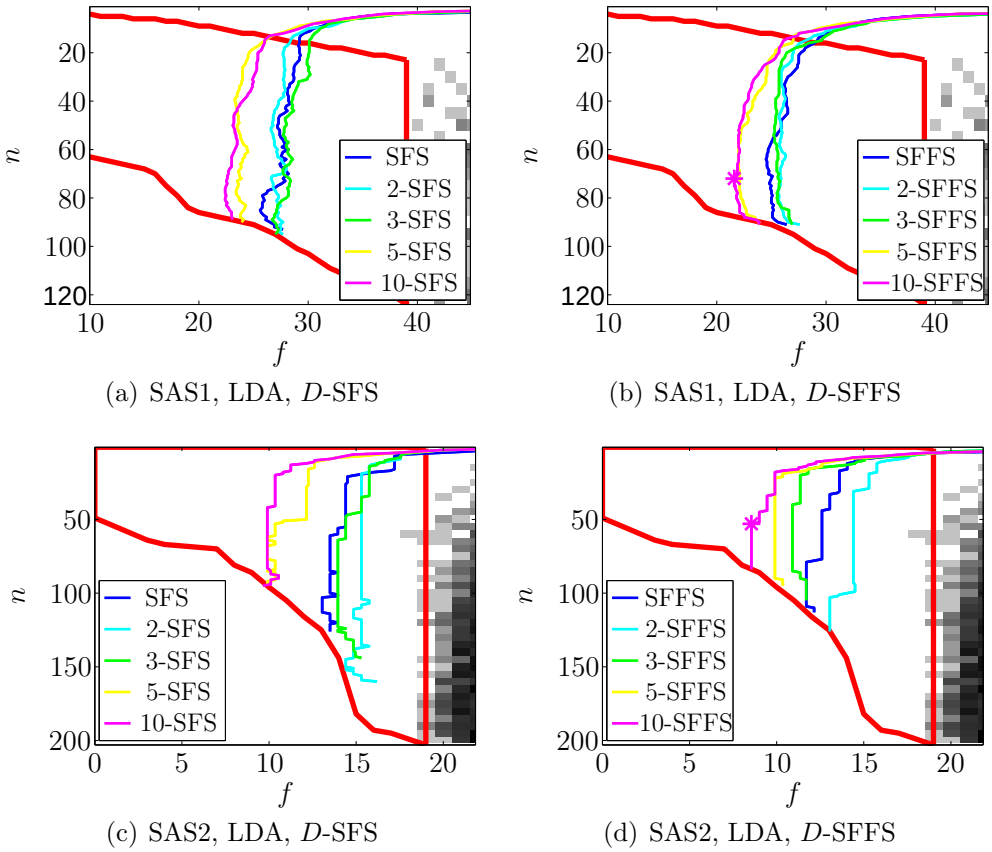
both databases, although its performance is not significantly better than the SVM for the SAS1 database and the k-NN for the SAS2 example. The reason for the good performance of the LDA and the bad performance of Mahalanobis' classifier is the covariance matrix estimation (see Secs. 3.2.2 and 3.2.3). While the former employs a single pooled covariance matrix, the latter computes a different covariance matrix for each class, which makes it more vulnerable to the curse of dimensionality. The LDA classifier has been adopted in the following for both data sets.

9.3 Feature Selection

The D-SFS and D-SFFS algorithms have been employed to estimate the optimal feature subset for both databases. In order to study the influence of the parameter D, the values $D = \{2, 3, 5, 10\}$ have been considered. For comparison, the standard SFS and SFFS are used. Results are included in Fig. 9.3. Each curve corresponds to a different value of D, and represents the value of f for the best n-feature subset, $\mathbf{t}_n^*$, $1 \leq n \leq n_{\mathrm{M}}$. For illustration, $\hat{p}_{f|n_i}$ has also been included and the most probable region for $\{f^*, n^*\}$ is again delimited by a red line. In order to focus on the area of interest, a significant change on the axes with respect to Fig. 9.2 has also been accomplished.

The maximum dimensionality n_{M} has been limited according to Eq. (3.26). For example, for the 5-SFFS applied to the SAS2 data set (see Fig. 9.3(d)), $q_{n|f}^{0.975} = 104$ for the smallest achieved value of the figure of merit, $f^* = 11$. Hence, the maximum number of iterations has been constrained to $n_{\mathrm{M}} = 104$.

Fig. 9.4 summarizes the results for the two data sets. The performance of both the D-SFS and D-SFFS algorithms degrades with respect to the standard SFS and SFFS



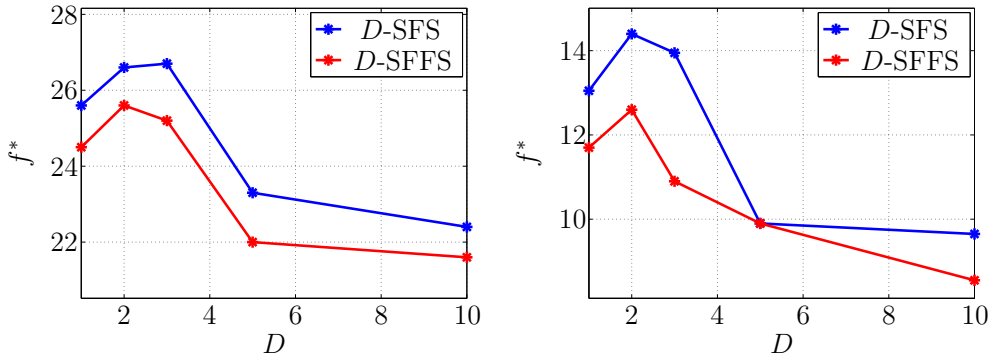
(a) SAS1, LDA, D-SFS (b) SAS1, LDA, D-SFFS

(c) SAS2, LDA, D-SFS (d) SAS2, LDA, D-SFFS

Figure 9.3: Feature selection results. The database, the classifier and the feature selection algorithm are indicated for each figure. The most probable region for $\{f^*, n^*\}$ is delimited by a red line. The distribution of f conditional on n, $10 \cdot \log(\hat{p}_{f|n_i})$ for $n_i = 1, \ldots, N$ is depicted in a logarithmic scale. Note the change of both axis limits and scale with respect to Fig. 9.2. Each figure shows the evolution of f as n increases for either the D-SFS or the D-SFFS algorithm for a certain value of D. As a reference, the standard SFS and SFFS algorithms have been used. The best result for each database is indicated by a star.

for $D = \{2, 3\}$. Despite the fact that increasing D leads, in most cases, to an improved performance, it is also possible that the algorithm chooses a combination of features that, although more convenient at a given step, results in a poorer final performance. For $D \geq 5$ though, a significant improvement is observed. The best results are obtained by the 10-SFFS for both examples. The 10-SFFS outperforms the standard SFFS by 12 % for the SAS1 data set and by 27 % for the SAS2 example. These results are only slightly better than those provided by the 5-SFFS algorithm, since the performance saturates for $D \gtrsim 5$. D-SFFS outperforms D-SFS for a given D, but the 5-SFS result is better than the standard SFFS one. It is, moreover, computationally more efficient.

The confusion matrices corresponding to the best f^* for the SAS1 and SAS2 data sets are included in Tables 9.3 and 9.4, respectively. The 10-SFFS algorithm provides the



(a) SAS1 (b) SAS2

Figure 9.4: D-SFS and D-SFFS performance. The results for $D = 1$ correspond to the standard SFS and SFFS algorithms.

Table 9.3: SAS1 confusion matrix. It corresponds to $\{f^*, n^*\} = \{21.6, 72\}$, reached by the 10-SFFS algorithm. The missed detected mines are indicated in red and the false alarms in blue.

	class 1	class 2	class 3
class 1	3478	17	109
class 2	4	240	64
class 3	6	9	114

optimal $f^* = 21.6$ at $n^* = 72$ for SAS1. Ten mines, i.e., 2.3 % of man made objects, are missed and 126 false alarms occur, which corresponds to 0.0022 false alarms per squared meter. Almost 17 % of the mines are assigned to a wrong mine class (cylinders classified as spheres or vice versa), however, this is not penalized for this data set ($\lambda' = 0$ in Eq. (9.3)). The SAS2 optimal $f^* = 8.5$ is obtained by the 10-SFFS algorithm at $n^* = 53$ (Table 9.4). No false alarm occurs and no mine is missed. For the SAS2 database, however, $\lambda' = 0.5$ and therefore, the selection of the correct mine class is considered. A total of 19 out of 185, that is, about 10 % of the man made objects, are assigned a wrong mine class. The optimal feature sets $\mathbf{t}^*$ for both databases are detailed in Tables 9.5 and 9.6. Both feature vectors include the segmentation overlap feature τ. The novel geometrical features o, $r_{\mathtt{sdw,hlt}}$, r_η and Ψ, whose measurement is based on either the Hough transform of the shadow skeleton or the accurate estimation of the shadow crossrange width (see Secs. 8.2 and 8.3), have also been selected.

Finally, Fig. 9.5 includes four snapshots of the SAS1 database where classification results are shown. the detected mines (both spheres and cylinders) are indicated in

Table 9.4: SAS2 confusion matrix. It corresponds to $\{f^*, n^*\} = \{8.5, 53\}$, reached by the 10-SFFS algorithm. The missed detected mines are highlighted in red and the false alarms in blue.

	class 1	class 2	class 3	class 4
class 1	3604	0	0	0
class 2	0	38	2	5
class 3	0	2	63	2
class 4	0	8	0	65

Table 9.5: Selected features for the SAS1 database. They produce the confusion matrix in Table 9.3.

Feature Type	**SAS1**
Statistical	$\xi'_{\texttt{hlt}}$, $\Delta\xi'_{\texttt{sdw,bkg}}$, $\Delta\xi_{\texttt{sdw,hlt}}$, $\Delta\xi_{\texttt{hlt,bkg}}$
Geo. `sdw`	χ, o, τ
Geo. `hlt`	$r_{\texttt{sdw,hlt}}$, $d_{\texttt{sdw,hlt}}$
Normalized Central Moments	$\bar{\sigma}_{2,0}$, $\bar{\sigma}_{3,0}$, $\bar{\sigma}_{0,5}$, $\bar{\sigma}_{3,2}$, $\bar{\sigma}_{2,3}$, $\bar{\sigma}_{7,0}$, $\bar{\sigma}_{5,2}$, $\bar{\sigma}_{3,4}$, $\bar{\sigma}_{4,3}$, $\bar{\sigma}_{7,1}$, $\bar{\sigma}_{9,0}$, $\bar{\sigma}_{1,8}$, $\bar{\sigma}_{8,1}$, $\bar{\sigma}_{2,7}$, $\bar{\sigma}_{7,2}$, $\bar{\sigma}_{3,6}$, $\bar{\sigma}_{6,3}$, $\bar{\sigma}_{5,4}$, $\bar{\sigma}_{1,9}$, $\bar{\sigma}_{9,1}$, $\bar{\sigma}_{2,8}$, $\bar{\sigma}_{8,2}$, $\bar{\sigma}_{3,7}$
Invariant M.	ι_5, ι_6, ι_7
Principal Components	ζ_4, ζ_{12}, ζ_{16}, ζ_{17}, ζ_{18}, ζ_{20}, ζ_{23}, ζ_{24}, ζ_{25}, ζ_{27}, ζ_{29}, ζ_{31}, ζ_{32}, ζ_{35}, ζ_{39}, ζ_{41}, ζ_{46}, ζ_{48}, ζ_{50}
2D-Fourier Coefficients	$\Xi_{0,5}$, $\Xi_{1,0}$, $\Xi_{1,4}$, $\Xi_{1,5}$, $\Xi_{2,0}$, $\Xi_{2,2}$, $\Xi_{2,4}$, $\Xi_{2,6}$, $\Xi_{3,0}$, $\Xi_{3,3}$, $\Xi_{4,4}$, $\Xi_{4,5}$, $\Xi_{5,0}$, $\Xi_{5,2}$, $\Xi_{5,5}$, $\Xi_{5,6}$, $\Xi_{6,1}$, $\Xi_{6,5}$

green, the false alarms in yellow and in red the missed detected mines. For the sake of clarity, we do not show the clutter that is classified as such. As expected, the false alarms are due to irregular parts of the seabed whose segmented shape is similar to either spheres or cylinders. Two main reasons account for the missed detected mines: either the segmentation is extremely poor, or the intensity of the shadow region is very light with respect to the other mines in the database. In this case, the statistical features cause the object to be classified as clutter.

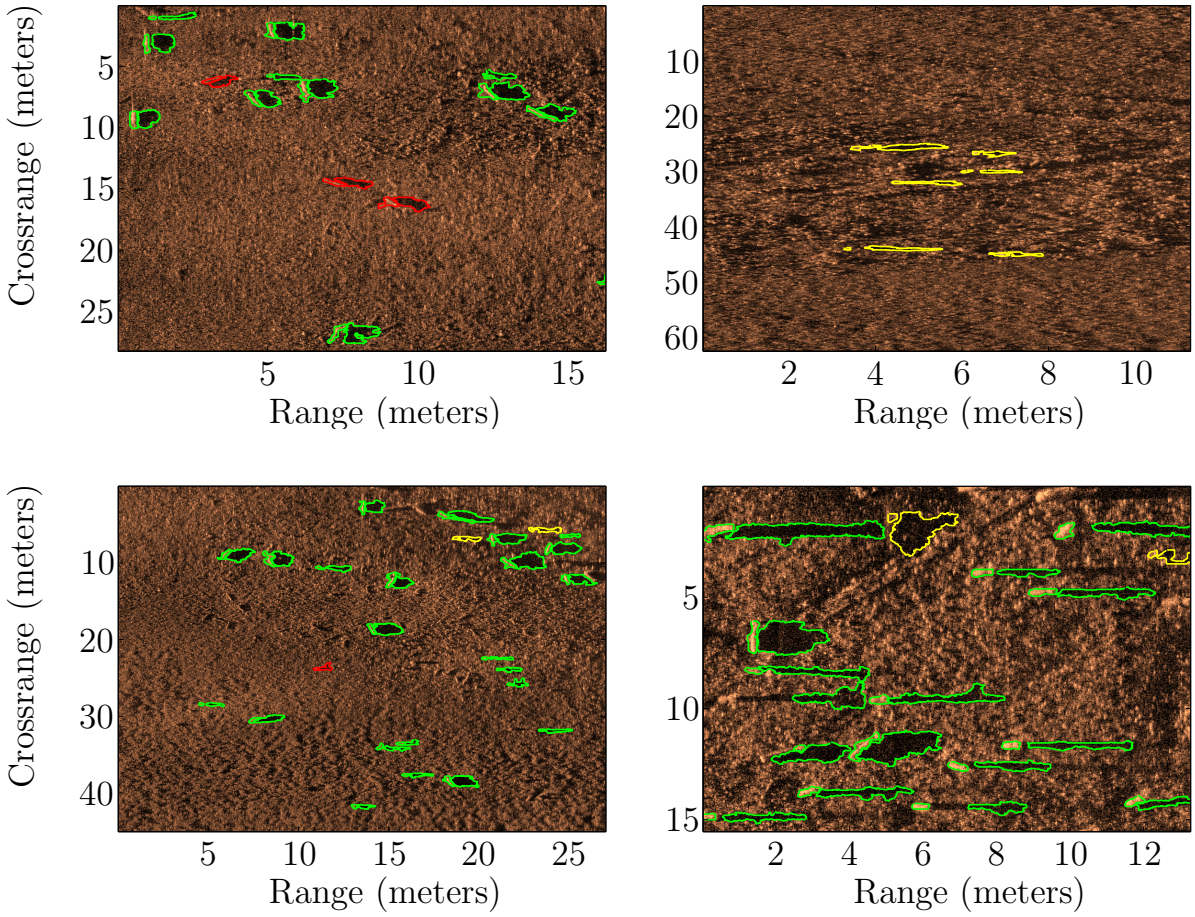


Figure 9.5: Illustration of the ADAC system output: the detected objects are indicated directly on the SAS images. The false alarms are marked in yellow, the detected mines in green and the missed detected mines in red.

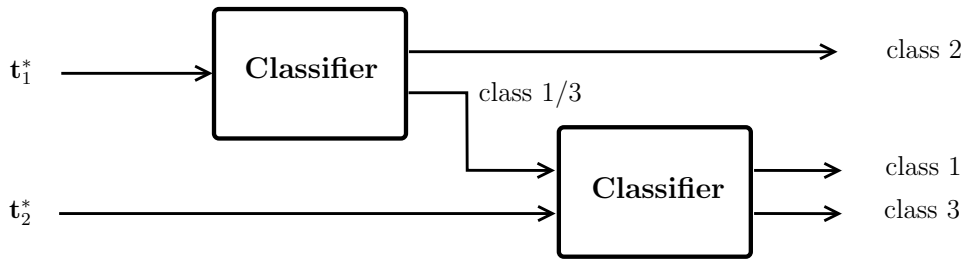


Figure 9.6: Cascade configuration classifier. A first classifier uses $\mathbf{t}_1^*$ to separate the spherical man made objects (class 2). Subsequently, a second classifier divides the remaining objects into cylinders (class 3) and clutter objects (class 1) according to $\mathbf{t}_2^*$.

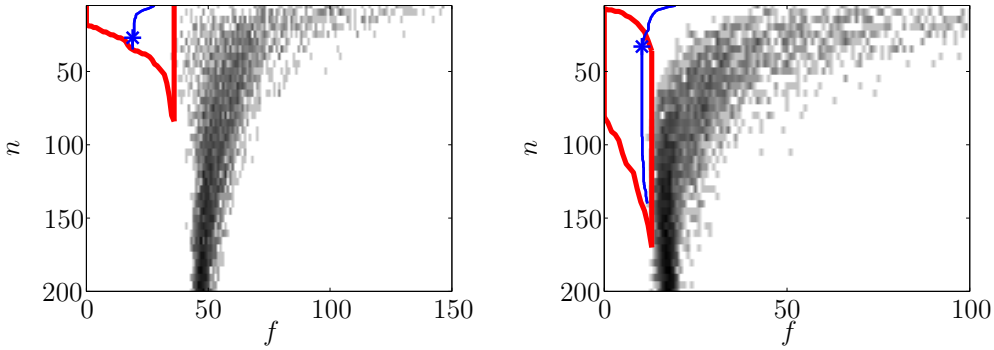
Table 9.6: Selected features for the SAS2 database. They produce the confusion matrix in Table 9.4.

Feature Type	SAS2
Statistical	$\xi'_{\texttt{bkg}}$, $\Delta\xi_{\texttt{sdw,bkg}}$
Geo. `sdw`	Γ, r_η, Ψ, τ
Geo. `hlt`	$r_{\texttt{hlt}}$, $d_{\texttt{sdw,hlt}}$, Δ_o
Normalized Central Moments	$\bar{\sigma}_{2,0}$, $\bar{\sigma}_{1,2}$, $\bar{\sigma}_{3,0}$, $\bar{\sigma}_{1,3}$, $\bar{\sigma}_{5,0}$, $\bar{\sigma}_{4,1}$, $\bar{\sigma}_{2,3}$, $\bar{\sigma}_{4,2}$, $\bar{\sigma}_{3,3}$, $\bar{\sigma}_{7,0}$, $\bar{\sigma}_{2,5}$, $\bar{\sigma}_{8,0}$, $\bar{\sigma}_{7,1}$, $\bar{\sigma}_{9,0}$
Invariant M.	ι_2
Principal Components	ζ_3, ζ_8, ζ_{13}, ζ_{18}, ζ_{19}, ζ_{23}, ζ_{26} , ζ_{32}, ζ_{33}, ζ_{35}, ζ_{36}, ζ_{37}, ζ_{38}, ζ_{39}, ζ_{45}, ζ_{46}
2D-Fourier Coefficients	$\Xi_{0,6}$, $\Xi_{1,0}$, $\Xi_{1,1}$, $\Xi_{1,3}$, $\Xi_{2,1}$, $\Xi_{2,2}$, $\Xi_{2,4}$, $\Xi_{3,3}$, $\Xi_{3,5}$, $\Xi_{4,6}$, $\Xi_{5,0}$, $\Xi_{5,3}$, $\Xi_{6,4}$

9.3.1 SAS1: Cascade Configuration Classifier

While the SAS2 classification results are outstanding, the false alarm rate of the SAS1 database leaves place for improvement. Note that most false alarms are due to the misclassification of clutter objects as cylinders (see Table 9.3). A cascade configuration of two binary classifiers is likely to provide a lower false alarm rate in the following manner. The first classifier distinguishes the spherical mines from the rest. The second classifier focuses on the differentiation of the cylinders from the clutter objects. Naturally, each classifier is based on a different feature set, $\mathbf{t}_1^*$ and $\mathbf{t}_2^*$. The scheme of the system is illustrated in Fig. 9.6.

In order to be able to compare the performance of this configuration with the standard scheme (single C-class classifier presented above), the LDA classifier has been adopted as well. First, the resampling method has been employed to estimate confidence intervals for the optimal number of features (see Fig. 9.7). Subsequently, the 10-SFFS algorithm estimates $\mathbf{t}_1^*$ and $\mathbf{t}_2^*$. For the figure of merit, $f := f_{\lambda,\lambda'}$ has been adopted, with $\lambda = 0.1$ and $\lambda' = 0$. By contrast with the single 3-class classifier, two classes are considered by each binary classifier and, therefore, the values of f cannot be directly compared. Hence, the comparison between both configurations is based on their final confusion matrices, included in Table 9.3 for the standard 3-class approach and in Table 9.7 for the cascade configuration classifier.



(a) First binary classifier (b) Second binary classifier

Figure 9.7: Cascade configuration classifier. The confidence intervals of the optimal number of features for the cascade configuration classifier are delimited by a red line. The feature selection results (in blue) correspond to the 10-SFFS applied to the SAS1 data set. The optimal figure of merit and optimal number of features has been indicated by a star. They correspond to the pairs $\{f^*, n^*\} = \{19, 27\}$ and $\{f^*, n^*\} = \{10.3, 33\}$ for the first and second binary classifiers, respectively.

Table 9.7: Confusion matrix of the SAS1 data set for the cascade configuration classifier.

	class 1	class 2	class 3
class 1	3494	52	58
class 2	13	295	21
class 3	5	0	103

As expected, the overall number of false alarms reduces. Instead of the 126 false alarms (0.0022 per squared meter) produced by the 3-class classifier, 110 (0.0019 false alarms per squared meter) occur for the cascade configuration. By contrast, the total number of missed detected mines increases from ten to 18 objects. The distribution of the false alarms and missed detection among the different mine classes, cylinders and spheres, is meaningful and, therefore, it is tackled in the following. The number of false alarms produced by clutter objects classified as cylinders significantly reduces, namely from 109 to 58 observations. The number of missed detected cylindrical mines diminishes from six to five. Furthermore, the number of cylindrical mines classified as spheres, although not considered by the figure of merit, drops from nine to zero. The number of spheres classified as cylinders reduces from 64 to 21 objects. On the contrary, the performance of this configuration regarding the spherical mines degrades with respect to the 3-class configuration: 52 instead of 17 clutter objects are classified as spheres, 13 instead of four spherical mines are missed detected. Hence, it can be concluded that

Table 9.8: Optimal feature sets for the cascade configuration classifier applied to the SAS1 database. They produce the confusion matrix in Table 9.7.

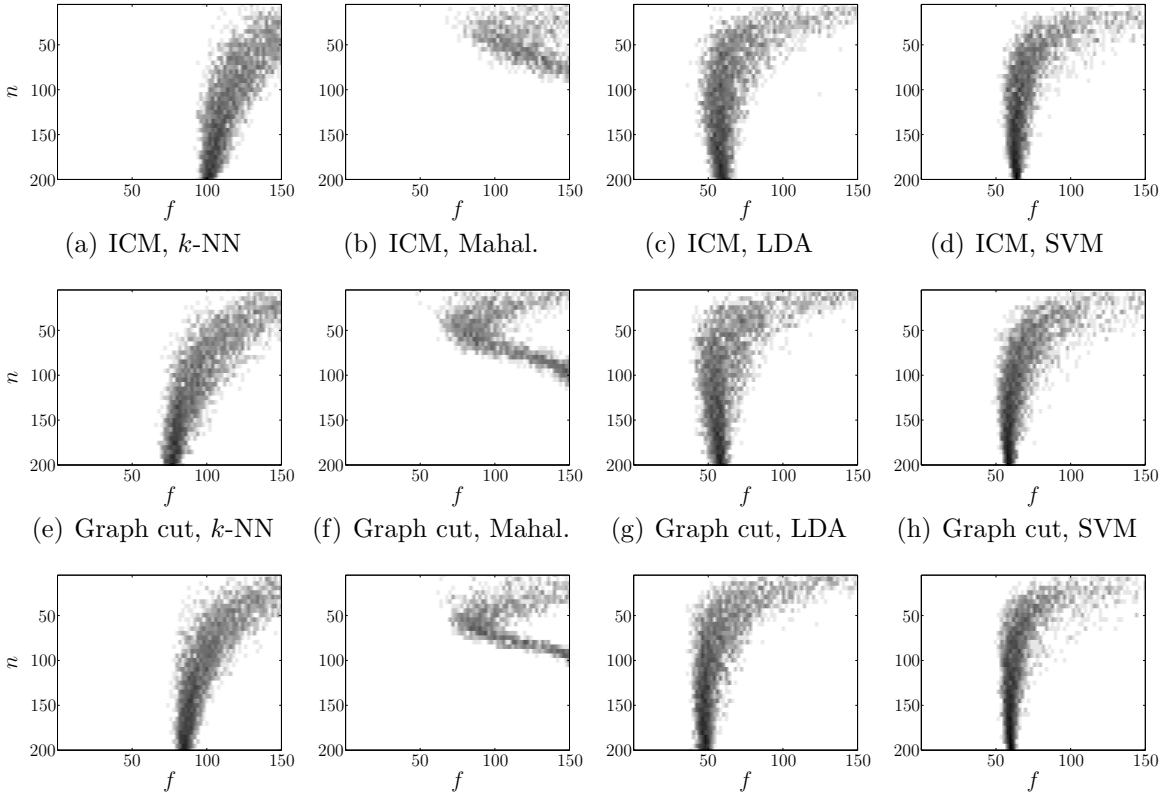
Feature Type	$\mathbf{t}_1^*$	$\mathbf{t}_2^*$
Statistical	$\xi'_{\texttt{bkg}}$, $\xi_{\texttt{sdw}}$, $\Delta\xi'_{\texttt{sdw,bkg}}$, $\Delta\xi_{\texttt{sdw,hlt}}$	$\xi'_{\texttt{bkg}}$, $\xi'_{\texttt{hlt}}$, $\Delta\xi'_{\texttt{sdw,bkg}}$, $\Delta\xi_{\texttt{sdw,hlt}}$
Geo. `sdw`	r_η	ρ, $r_{\texttt{sdw}}$, τ, η_v
Geo. `hlt`	$r_{\texttt{hlt}}$, $\Pi_{\texttt{hlt}}$, $r_{\texttt{sdw,hlt}}$	$\gamma_{\texttt{hlt}}$, Δ_o, $d_{\texttt{sdw,hlt}}$
Normalized Central Moments	$\bar{\sigma}_{1,2}$, $\bar{\sigma}_{2,4}$, $\bar{\sigma}_{1,6}$, $\bar{\sigma}_{0,9}$, $\bar{\sigma}_{8,1}$, $\bar{\sigma}_{10,0}$	$\bar{\sigma}_{8,0}$, $\bar{\sigma}_{7,1}$
Invariant M.	ι_2, ι_7	
Principal Components	ζ_{27}, ζ_{29}, ζ_{50}	ζ_2, ζ_5, ζ_6, ζ_8, ζ_{11}, ζ_{13}, ζ_{16}, ζ_{30}, ζ_{32}, ζ_{41}
2D-Fourier Coefficients	$\Xi_{0,2}$, $\Xi_{0,5}$, $\Xi_{0,7}$, $\Xi_{1,3}$, $\Xi_{2,3}$, $\Xi_{3,0}$, $\Xi_{3,4}$, $\Xi_{5,5}$, $\Xi_{6,1}$	$\Xi_{0,2}$, $\Xi_{1,5}$, $\Xi_{2,2}$, $\Xi_{2,3}$, $\Xi_{2,4}$, $\Xi_{4,4}$, $\Xi_{4,6}$, $\Xi_{5,4}$, $\Xi_{6,2}$, $\Xi_{6,3}$

the cascade configuration yields better results than the standard 3-class classifier for cylindrical objects but, on the other hand, it degrades with respect to the spherical object class. As a whole, the cascade configuration results into a lower false alarm rate but a higher missed detection rate than the 3-class approach. The correct mine class is more often selected by the cascade configuration, that is, less spheres are classified as cylinders and vice versa.

The elements in $\mathbf{t}_1^*$ and $\mathbf{t}_2^*$ are specified in Table 9.8. Although they do not coincide in almost any feature, both sets include several novel geometrical features based on both the shadow skeleton and the accurate estimation of the shadow crossrange width (see Secs. 8.2 and 8.3). While $\mathbf{t}_1^*$ consists of 27 elements, $\mathbf{t}_2^*$ has 33 features. Both dimensionalities are significantly lower than the size of $\mathbf{t}^*$ for the 3-class classifier, which makes each of the binary classifiers computationally less demanding.

9.4 Segmentation Comparison

All classification results presented above have been obtained utilizing the features extracted from the graph cut segmentation results. In this section, these classification results are compared with those provided by the other two segmentation algorithms presented in Chapter 7, the ICM and the AC algorithms.



(a) ICM, k-NN (b) ICM, Mahal. (c) ICM, LDA (d) ICM, SVM

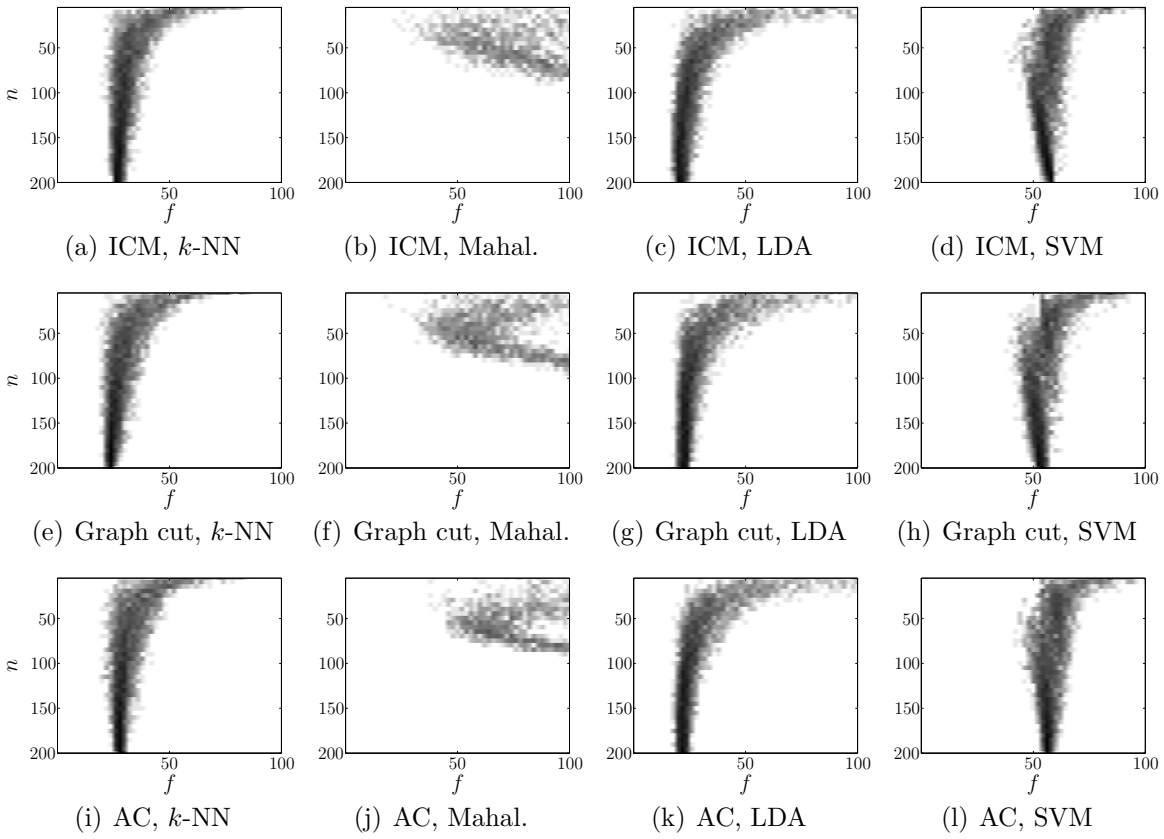
(e) Graph cut, k-NN (f) Graph cut, Mahal. (g) Graph cut, LDA (h) Graph cut, SVM

(i) AC, k-NN (j) AC, Mahal. (k) AC, LDA (l) AC, SVM

Figure 9.8: Comparison of segmentation performance for the SAS1 data set. The curves represent $10 \cdot \log(\hat{p}_{f|n_i})$ for $n_i = 1, \ldots, N$, estimated by histogram techniques from the figure of merit estimates. The scale is common for all figures and spans between -25 dB (white) and -5 dB (black).

Instead of fixing a classification system and comparing, e.g., the confusion matrix stemming from each segmentation algorithm, the quality assessment for classifier performance Q proposed in Chapter 3 (see Eq. (3.20)) can be exploited. This comparison is more meaningful, since it is not constrained to a specific feature set but it regards the quality of the whole feature space subject to a segmentation method and a classification system.

The distribution of the figure of merit subject to the number of features, $\hat{p}_{f|n_i}$, $n_i \in \{1, \ldots, N\}$, for all four classification systems and all three segmentation methods has been estimated by histogram techniques. They are depicted in Fig. 9.8 for the SAS1 database and in Fig. 9.9 for the SAS2 data set (note that the graph cut curves correspond to those included in Fig. 9.2). The shape of $\hat{p}_{f|n_i}$ is determined by the classification system rather than the segmentation algorithm. Given a classification system, however, the energy of $\hat{p}_{f|n_i}$ might be shifted more or less towards lower values of f for a certain segmentation method. For instance, for the k-NN classifier, the



(a) ICM, k-NN (b) ICM, Mahal. (c) ICM, LDA (d) ICM, SVM

(e) Graph cut, k-NN (f) Graph cut, Mahal. (g) Graph cut, LDA (h) Graph cut, SVM

(i) AC, k-NN (j) AC, Mahal. (k) AC, LDA (l) AC, SVM

Figure 9.9: Comparison of segmentation performance for the SAS2 data set. The curves represent $10 \cdot \log(\hat{p}_{f|n_i})$ for $n_i = 1, \ldots, N$, estimated by histogram techniques from the figure of merit estimates. The scale is common for all figures and spans between -25 dB (white) and -5 dB (black).

energy of $\hat{p}_{f|n_i}$ is placed at higher f values for the ICM than for the graph cut and AC segmentation methods (see Figs. 9.8(a), 9.8(e) and 9.8(i)).

The values of $\hat{p}_{f|n_i}$ have been employed to calculate the quality measure Q. They are included in Tables 9.9 and 9.10 for the SAS1 and SAS2 databases, respectively. In order to visualize the comparison of the different methods, the data in the tables are represented in Figs. 9.10 and 9.11. The former displays a curve for each classification method, allowing for a comparison of the classification performance for each given segmentation algorithm. The latter includes a curve for each segmentation method, which provides a comparison of the results yielding from each classification system by all three segmentation methods.

From the fact that the curves in Fig. 9.10 do not cross each other, it stems that, given a database, the ranking of classification systems is identical for all segmentation algorithms. However, the ranking is different for each data set. The LDA performs best

	ICM	Graph cut	AC
k-NN	0.008	0.010	0.010
Mahal.	0.006	0.007	0.007
LDA	0.015	0.016	0.018
SVM	0.014	0.014	0.015

Table 9.9: Performance assessment Q, with $\psi = 1$, for the SAS1 database.

	ICM	Graph cut	AC
k-NN	0.033	0.035	0.033
Mahal.	0.024	0.024	0.021
LDA	0.039	0.037	0.038
SVM	0.018	0.019	0.017

Table 9.10: Performance assessment Q, with $\psi = 1$, for the SAS2 database.

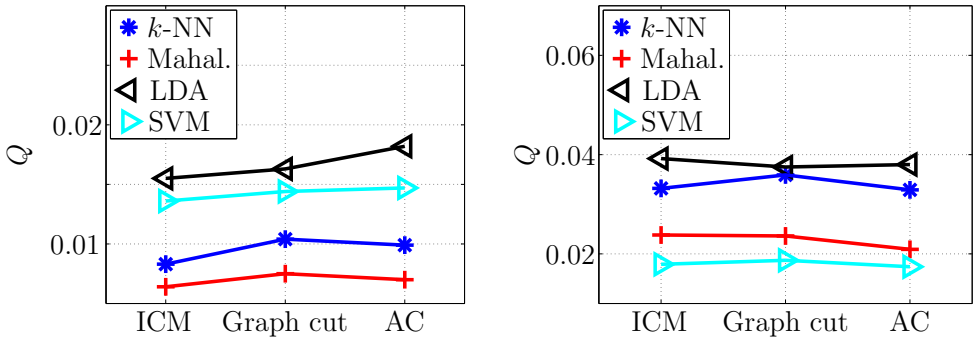
for both SAS1 and SAS2, but the other three classification systems are sorted in different order. Thus, the worst classifier candidate for the SAS1 data set is Mahalanobis' classifier, and the SVM for the SAS2 database.

On the other hand, the curves in Fig. 9.11 cross, which implies that the values of Q provided by different segmentation methods given a classification system, might have a different ranking for different classifiers. For the SAS2 data set, the max-flow/min-cut algorithm yields the best performance for all classifiers except the LDA. For the LDA classifier, the ICM performs slightly better. For the SAS1 database, the graph cut algorithm yields the best results for the k-NN and Mahalanobis' classifiers, while the AC algorithm performs best when either the LDA or the SVM are chosen. Therefore, another possible improvement for the SAS1 classification results could consist of using AC instead of min-cut/max-flow as segmentation algorithm.

It reveals that the classifier has a stronger influence on the performance of the complete system than the segmentation method. Very likely, this is due to the fact that all three compared segmentation algorithms are, indeed, rather sophisticated. Furthermore, the suboptimal segmentation provided by the ICM approach does not have a strong influence on the final result, since the features have been specially designed to remain insensitive to such scenarios.

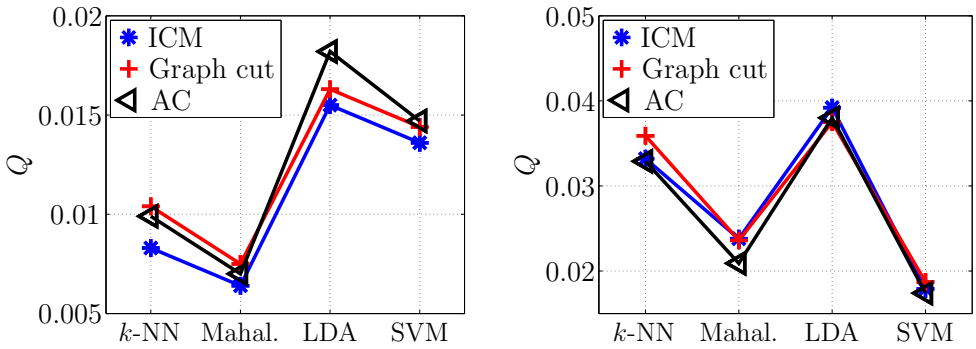
9.5 Computational Cost

In this section, the computational costs of the resampling algorithm and the feature selection algorithms are discussed. All time estimations have been calculated with a computer equipped with an Intel i5 4 core 2.8 GHz processor. All programs have been written in Matlab [130].



(a) SAS1 (b) SAS2

Figure 9.10: Comparison of the performance of the different segmentation algorithms with the different classification methods.



(a) SAS1 (b) SAS2

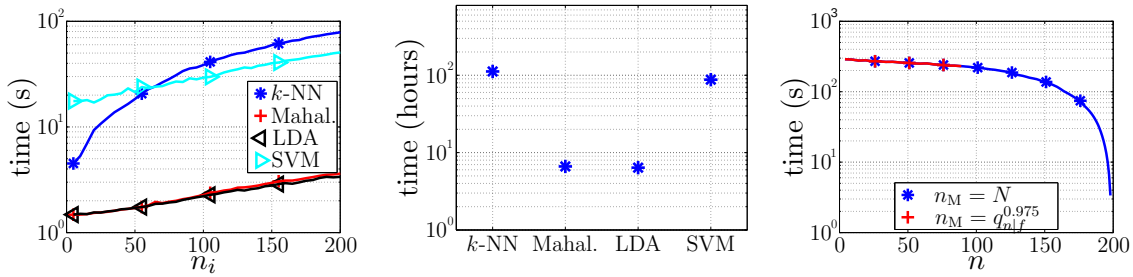
Figure 9.11: Comparison of the performance of the different segmentation algorithms with the different classification methods.

9.5.1 Resampling Method

The resampling method proposed in Chapter 3 is computationally intensive. As described in Sec. 3.4, for each n_i, $n_i \in \{n_1, n_2, \ldots, n_{\max}\}$, the figure of merit is calculated for each sample $\mathbf{t}'_b$, $1 \leq b \leq N_B$.

Computing the value of f for a certain $\mathbf{t}'_b$ requires classifying the whole database. This time consumption depends on the number of observations in the database, the feature set dimensionality n_i, the classification system, and the method employed to divide the database into training and test set (e.g., 5-fold cross validation).

Fig. 9.12(a) depicts the computational cost of obtaining the figure of merit for a single sample as a function of n_i for the four classification systems considered in this thesis.



(a) Cost of computing f for a single sample as a function of n_i for the four classification systems.

(b) Cost of computing f for all samples required to obtain $\hat{p}_{f|n_i}$ for the four classification systems.

(c) Computation cost of an SFS iteration for the LDA classifier. If n_M is constrained by $q^{0.975}_{n|f}$, the number of iterations reduces to less than half.

Figure 9.12: Computational cost of the resampling method and the SFS algorithm for the SAS1 database with 5-fold cross validation.

The SAS1 data set has been employed and the 5-fold cross validation has been adopted. Mahalanobis' and LDA classifiers require at most a couple of seconds to compute f, even for the highest n_i. By contrast, the k-NN classifier needs more than a minute for $n_i > 150$. Such high computational time might seem to contradict the simplicity of the k-NN classifier (see Sec. 3.2.1). However, the calculation of the Euclidean distance is computationally expensive, specially as the dimensionality increases. Moreover, it has to be calculated for all observations in the database, whose size is big due to the amount of clutter objects. The SVM computational time is also high, in this case due not only to the size of the data set but also to the complexity of the classifier (see Sec. 3.2.4).

The cost of computing all figure of merit estimates f'_b required for obtaining $\hat{p}_{f|n_i}$ for $n_i \in \{1, 5, 10, \ldots, 200\}$ and $N_B = 100$ is illustrated in Fig. 9.12(b) for the SAS1 database. Note that about a hundred hours are needed by the k-NN and the SVM classification systems. Since the resampling method is to be applied offline during the design process of the ADAC system, this computational cost is affordable. Furthermore, the calculation of the figure of merit estimates associated to the different samples can easily be parallelized.

9.5.2 Feature Selection

In this section, the computational cost of the SFS and SFFS algorithms is investigated. The D-SFS algorithm consumes D times more time than the SFS, and the D-SFFS computational cost is about D times the computational cost of the SFFS algorithm.

At iteration n, the SFS algorithm needs to compute the figure of merit of the $N-n$ feature subsets that are candidates to become $\mathbf{t}_n^*$. For small values of n, the feature subset dimensionality is small and therefore the calculation of f is faster (see Fig. 9.12(a)) but, at the same time, more candidates are to be considered. The time required by the SFS algorithm as a function of n is illustrated in Fig. 9.12(c). Since the quality assessment establishes that the LDA is the best classification system for both SAS1 and SAS2, only this classifier is considered. The total cost of the SFS algorithm is calculated by integrating the curve. For the SAS1 data set, it is about 26 hours if $n_{\mathrm{M}} = N$. However, if n_{M} is constrained by the $q_{n|f}^{0.975}$ confidence interval (as depicted in Fig. 9.3(a)), then $n_{\mathrm{M}} = 90$ and the time required by the SFS algorithm reduces to 15 hours.

The computational cost of the SFFS algorithm depends on how often features are removed from the optimal feature set. In general, it is considered that a typical SFFS realization is between six and ten times slower than the SFS algorithm for the same database and classification system.

Since a working ADAC system simply classifies the objects for a given classification system and feature subset, the computational cost of the classification is rather low. For instance, the cost of classifying all objects in the SAS1 data set for the LDA classifier and a feature subset of 72 elements (as chosen in Sec. 9.3) is approximately 2 seconds (see Fig. 9.12(a)).

Chapter 10
Conclusions and Future Work

In the second part of this thesis the problem of Automatic Detection And Classification (ADAC) of underwater objects for mine hunting applications has been addressed. A processing chain consisting of segmentation, feature extraction and classification has been proposed. The selection of the classification system among a set of candidates and the selection of the optimal feature subset have been performed according to the algorithms proposed in Part I. They have been tested on two extensive data sets of Synthetic Aperture Sonar (SAS) images, SAS1 and SAS2, containing both mines and clutter objects.

A summary and the main conclusions are provided in Sec. 10.1. Finally, Sec. 10.2 provides an outlook for possible future work.

10.1 Conclusions

10.1.1 SAS Image Segmentation

The ADAC chain starts with the detection of the objects in the scene, which is accomplished by segmenting the image. Three regions are considered, the shadow of the objects, their highlight and the background.

Three algorithms for segmentation of SAS images have been investigated. The Iterative Conditional Modes (ICM) and the min-cut/max-flow algorithms are based on a Markov Random Fields (MRF) model of the image of interest. The Active Contours (AC) algorithm is a contour fitting approach. Special attention has been paid to the initialization of the three algorithms, since it is a crucial matter.

The MRF image representation combines a model of the image intensity, the likelihood function, and a model of the pixel neighbor relations, the Markovian *a priori* probability. According to this model, the optimal segmentation of the image maximizes the *a posteriori* probability conditional on the image. Finding its absolute maximum is computationally prohibitive. The ICM and min-cut/max-flow methods approach it by a local maximum.

The ICM segmentation method requires an estimation of the likelihood function and the Markovian probability. For this purpose, the ICE algorithm is employed, which requires a segmentation initialization. A non-parametric approach has been proposed for the likelihood function, allowing for a more accurate and robust pdf estimation than the traditional Weibull and Rayleigh models. Three initialization schemes have been investigated and compared with a well-established approach. The enhanced initialization scheme, proposed in this thesis, is an unsupervised method that provides significantly better results for 80 % of the man made objects in the SAS1 data set.

The min-cut/max-flow algorithm has been applied for segmentation of sonar images in the context of this thesis for the first time. A graph representation of the image is adopted, where each node corresponds to a pixel and the edges between nodes model the so-called regional and boundary properties of the image. The former are related to the image intensity likelihood function and the latter refer to the pixel neighbor relations of the MRF image model. The min-cut/max-flow algorithm divides the graph nodes into two groups according to the edge properties. It has been found that the segmentation result is rather insensitive to variations of the edge properties, as long as they keep the correct tendency. By contrast, the initialization of the algorithm has a significant impact on the result. A novel initialization scheme, which is based on the ICM segmentation result, is proposed. The performance of the min-cut/max-flow algorithm is influenced by two parameters. The first one controls the impact of the initialization on the final segmentation and the second decides the relative weight of regional and boundary properties. A thorough parameter study has been accomplished in order to find suitable values for the application at hand.

The AC algorithm has also been tested. A closed curve is deformed in order to minimize a cost function, whose absolute minimum theoretically coincides with the edge between regions. The cost function is based on the likelihood function of the image. Unlike gradient based AC implementations, it successfully handles the noisy nature of sonar images. The algorithm has a tendency to converge to local minima. A novel solution to this issue has been proposed. The AC initialization is based on the ICM segmentation result as well. The contour is initialized as a rectangle around the center of mass of the ICM segmented region.

The segmentation results provided by all algorithms have been compared for a set of cylindrical, spherical and clutter underwater objects of the SAS1 database. By eye inspection, one can see that the min-cut/max-flow and AC algorithms are more insensitive to irregular backgrounds than the ICM method, providing more regular region shapes of the man made objects. This is mainly due to their more accurate

initialization. Therefore, the min-cut/max-flow and AC methods are likely to yield better classification results.

Finally, the computational cost has been investigated. For the examples at hand, the computational times of the three algorithms have the same order of magnitude, between 10 and 100 seconds for an Intel i5 4 core 2.8 GHz processor. Since the AC and max-flow/min-cut algorithms require the ICM result though, these two methods are in practice roughly twice as slow as the ICM approach.

10.1.2 Feature Extraction

Each segmented object is characterized by a set of descriptors. Several kinds have been investigated. On the one hand, statistical features are regarded. They are based on a Weibull parametric model of the intensity of the pixels belonging to the different regions (shadow, highlight and background). They exploit the fact that the pixel intensity of objects belonging to different classes follow different distributions.

On the other hand, shape descriptors characterize both the shadow and the highlight regions. The shadow descriptors take distinct values for the spherical objects, whose shadow exhibits an invariantly elongated shape. By contrast, cylindrical objects are better characterized by the features of their highlight. They take into account not only the highlight shape but also its relative position with respect to the shadow.

For the shadow, some standard sets of descriptors are considered: normalized central moments, invariant moments, principal components and 2D-Fourier coefficients. Moreover, a group of novel descriptors is proposed. They focus on the reduction of the feature variability as the orientation of the object with respect to the sonar system changes, or in poor segmentation scenarios. For this reason, they are specially valuable if the ICM segmentation is employed. The topological skeleton of the shadow has been used in order to achieve a correct estimation of several of these features. A novel feature that estimates the segmentation reliability is especially useful for characterizing clutter objects.

The computational cost of the feature extraction is negligible compared with the segmentation cost.

10.1.3 Classification and Feature Selection

The general methods for ADAC system design proposed in Chapters 3 (selection of the optimal classification system) and 4 (selection of the optimal feature subset) have been applied to the specific application of mine hunting for the two databases at hand, SAS1 and SAS2.

In order to quantify the ADAC system performance, a figure of merit is required. Instead of the traditional overall misclassification rate, a novel figure of merit is defined. It takes into account the class imbalance of the SAS databases, allowing for minimizing the missed detected mines despite the dominance in size of the clutter class. The misclassification of mines as clutter (missed detection) is more critical than assigning them a wrong mine class. This issue is also regarded by the proposed figure of merit.

The quality assessment for classifier performance has been applied for comparing four classification systems: the k-Nearest Neighbor (k-NN), Mahalanobis', Linear Discriminant Analysis (LDA) and Support Vector Machines (SVM). The LDA yields the best results for both data sets. While the SVM performance is also good for the SAS1 database, it is poor for the SAS2 example. By contrast, the k-NN classifier yields good classification results for the SAS2 database, but they are poor when applied to the SAS1 data set. Mahalanobis' classifier suffers a strong peaking effect for both data sets. This is due to the small number of observations of the mine classes, which results into a poor estimation of the class covariance matrix. Since the LDA classifier employs the pooled covariance matrix, the effect of the curse of dimensionality is significantly weaker.

Applying the D-SFS and D-SFFS algorithms, $D = \{2, 3, 5, 10\}$, the optimal feature set for each database has been calculated. The best results corresponds to the 10-SFFS algorithm for both data sets. The 10-SFFS method is also the computationally most expensive one. The classification results are excellent for the SAS2 database, neither mines are missed detected nor false alarms happen. Only 10 % of the mines are assigned a wrong mine class. The SAS1 performance is good but less outstanding: 2.3 % of the mines are missed detected, and 0.0022 false alarms per square meter occur. The optimal feature set for both databases comprise numerous of the descriptors novel to this thesis, such as the shadow skeleton based attributes and the feature characterizing the segmentation reliability.

In order to reduce the false alarm rate for the SAS1 data set, a cascade configuration classifier has been regarded. A first binary classifier discerns the spherical mines. A

second one separates the clutter from the cylindrical objects. The number of false alarms due to clutter objects classified as cylindrical mines has been reduced by more than 45 %, and less mines are assigned a wrong mine class. However, the number of missed detected mines increases by eight objects.

Although the three segmentation methods, ICM, min-cut/max-flow and AC, have been compared by eye inspection of their results, it is their classification performance what matters. Therefore, the resampling based quality assessment has been used to compare the three segmentation algorithms for the four available classification systems, k-NN, Mahalanobis', LDA and SVM. It has been found that the ranking among segmentation methods depends on the database on the one hand, and on the classification system on the other hand. The min-cut/max-flow algorithm is optimal for two out of four classifiers for the SAS1 database, and for three if the SAS2 data set is considered. The ICM algorithm provides the worst performance for all classifiers if the SAS1 database is employed. For the SAS2 data set, however, the ICM is the worst segmentation method only combined with the k-NN classifier. However, the variation of the quality measure as a function of the segmentation algorithm is much smaller than as a function of the classification method, which implies that the performance of the overall system is more strongly determined by the classification system than by the segmentation algorithm, at least for the methods considered in this thesis. Presumably, this is due to the fact that all three segmentation algorithms are, indeed, rather sophisticated. Moreover, the irregular segmented regions provided by the ICM segmentation method are neutralized by the accuracy of the feature extraction.

Finally, the computational cost of the methods has been considered. Both the resampling method and the feature selection algorithms are computationally expensive. Since the algorithms are employed offline for the design of the system, such high computational cost is not crucial. Still, the number of iterations required by the D-SFS and the D-SFFS is drastically reduced by the confidence intervals for optimal number of features provided by the resampling algorithm.

For the working ADAC system, which simply classifies the objects for a single classification system and feature subset, the computational cost of the classification is negligible with respect to the cost of the segmentation.

10.2 Future Work

In the ADAC chain consisting of segmentation, feature extraction and classification, it is the segmentation part that requires more computational time, between 10 to

100 seconds per object. For a real time application, such computational time is not acceptable. A straightforward way to speed up the algorithms consists of employing a faster programming language than Matlab, such as C or Fortran. In fact, a C implementation of the min-cut/max-flow algorithm has already been tested. In average, it requires 0.6 seconds per object, which is reasonably fast.

The comparison of the proposed feature based ADAC system with some template fitting methods for mine hunting (e.g. [81–83,97]) is desirable. Given the independence of both approaches, it is indeed likely that their combination could produce improved and more reliable results.

Employing multiple views of the same object or interferometry (3D images of the seabed) could allow for new meaningful features that, very probably, would result in a better performance.

Finally, although the SAS databases contain a fair amount of mines, it is desirable to test the system with bigger databases and different kinds of backgrounds.

Appendix

Feature Distributions

The estimated distribution of the features presented in Chapter 8 are illustrated. They have been extracted from the ICM segmentation result of the SAS images in the SAS1 database. The clutter objects correspond to class 1 (red), while the spherical and cylindrical man made objects are assigned to classes 2 (blue) and 3 (black), respectively. A KDE, with the bandwidth optimized for Gaussian distributions, has been employed to estimate the pdf, $\hat{p}$, from the data. For features referring to length measures, such as the width of the shadow or the length of the highlight, pixels instead of meters are used as measure unit.

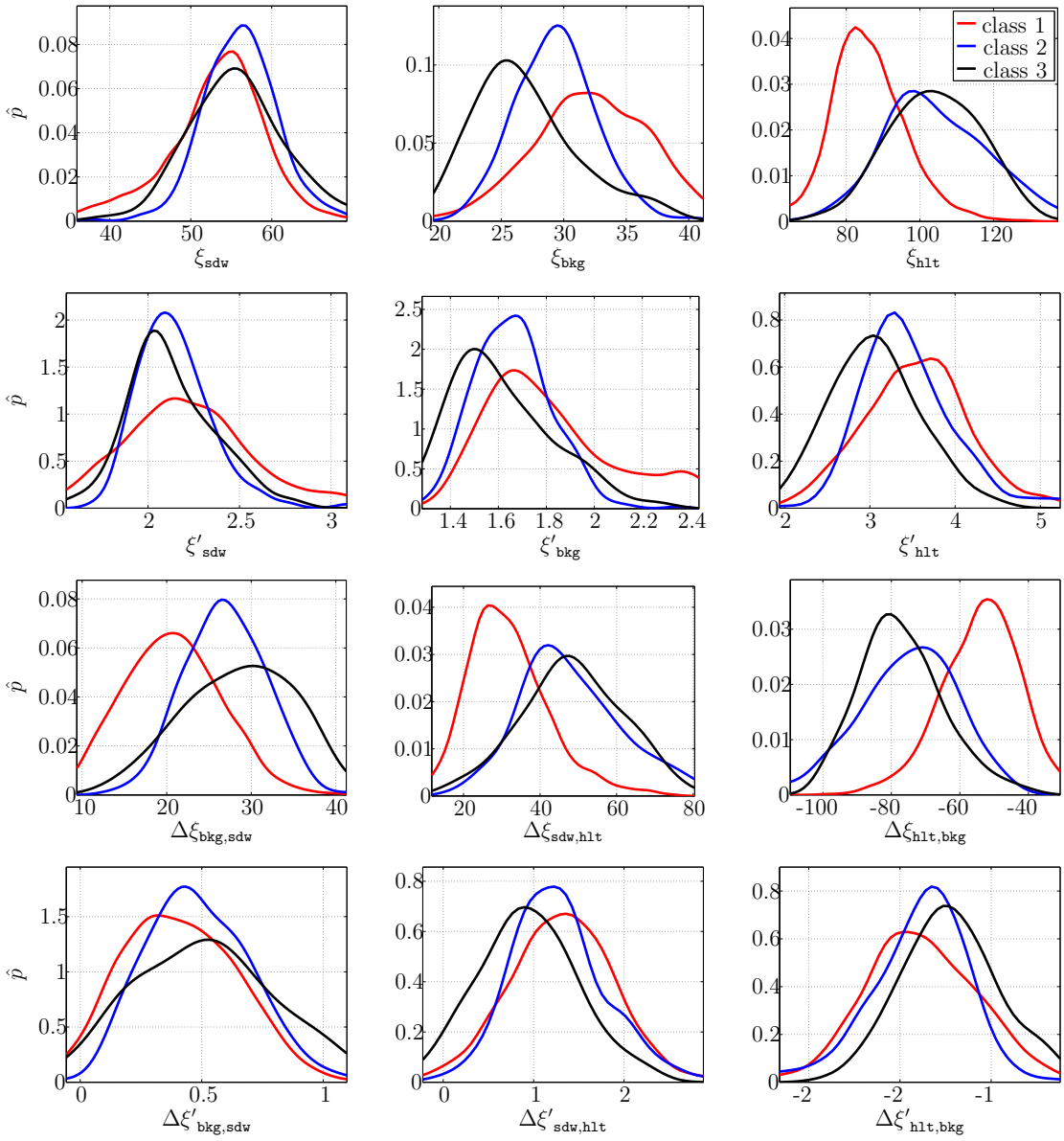


Figure A.1: Statistical features.

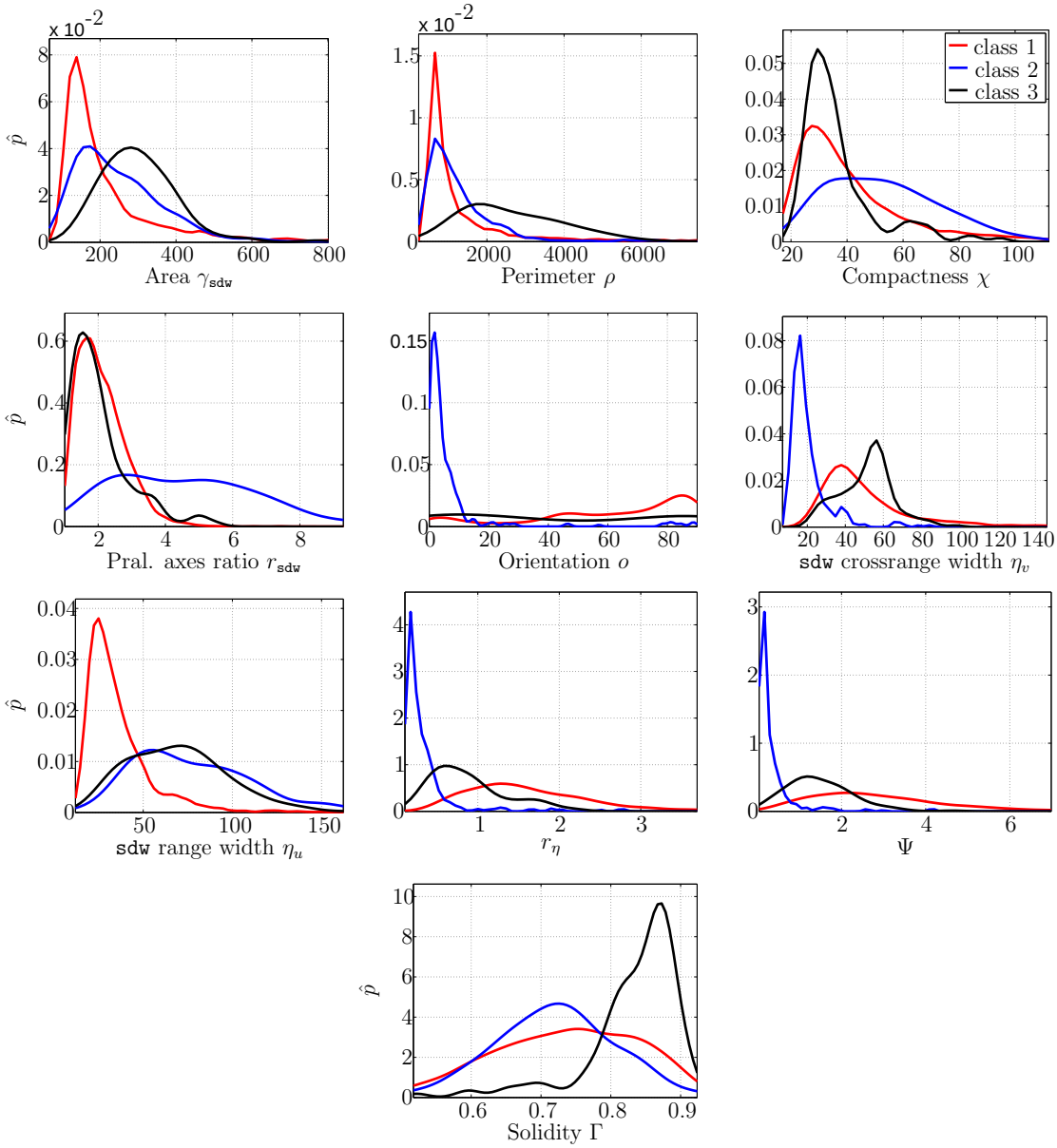


Figure A.2: Shadow geometrical features

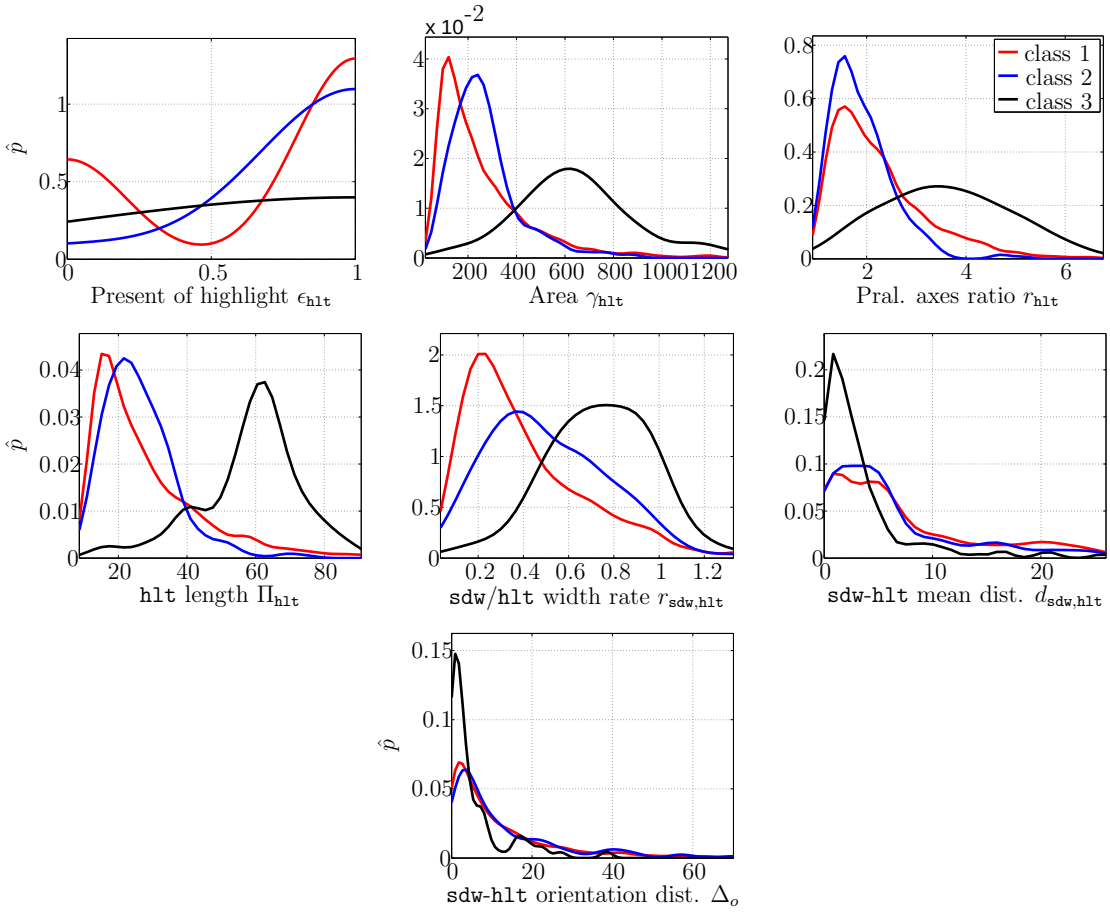


Figure A.3: Highlight geometrical features

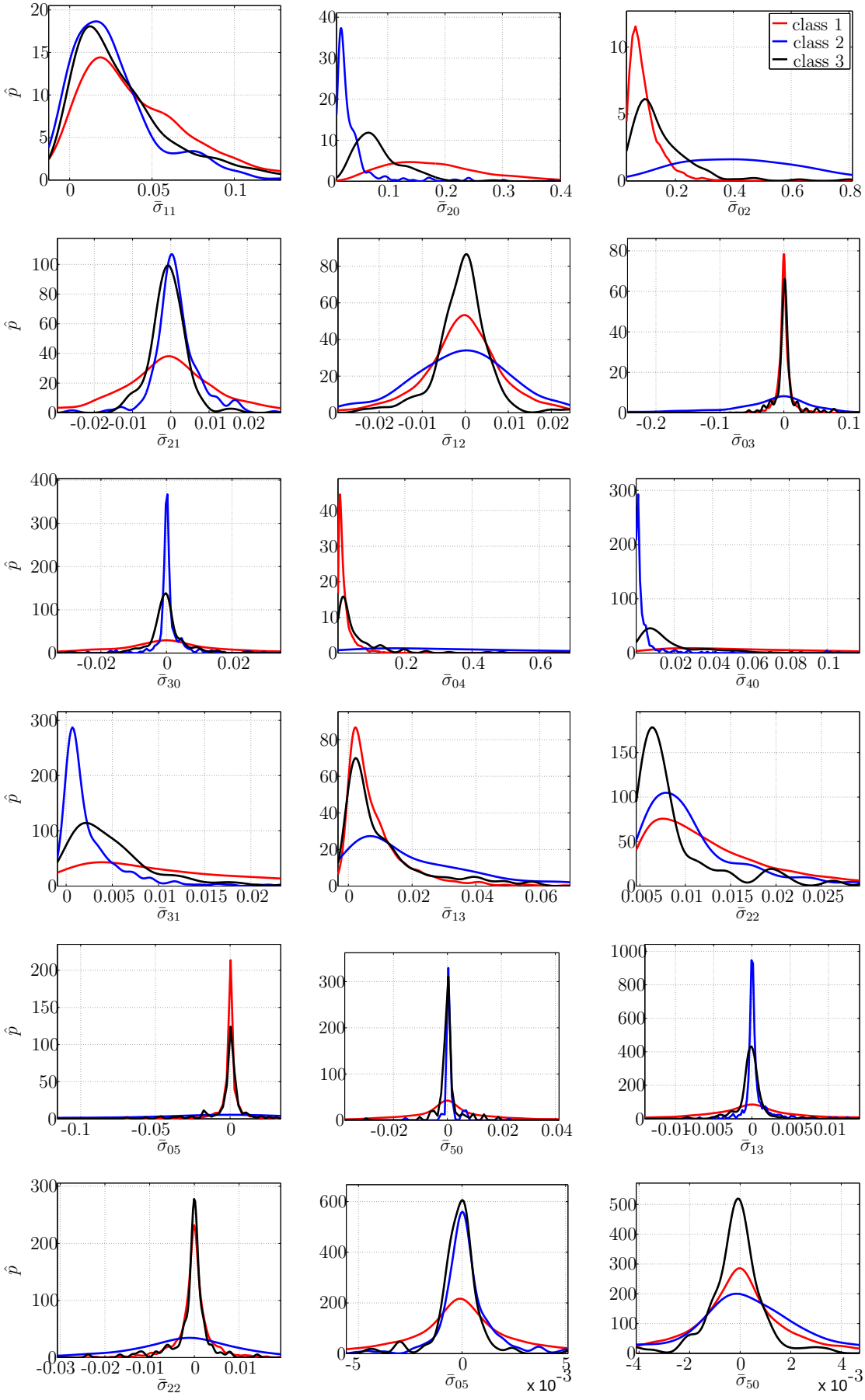


Figure A.4: Normalized central moments

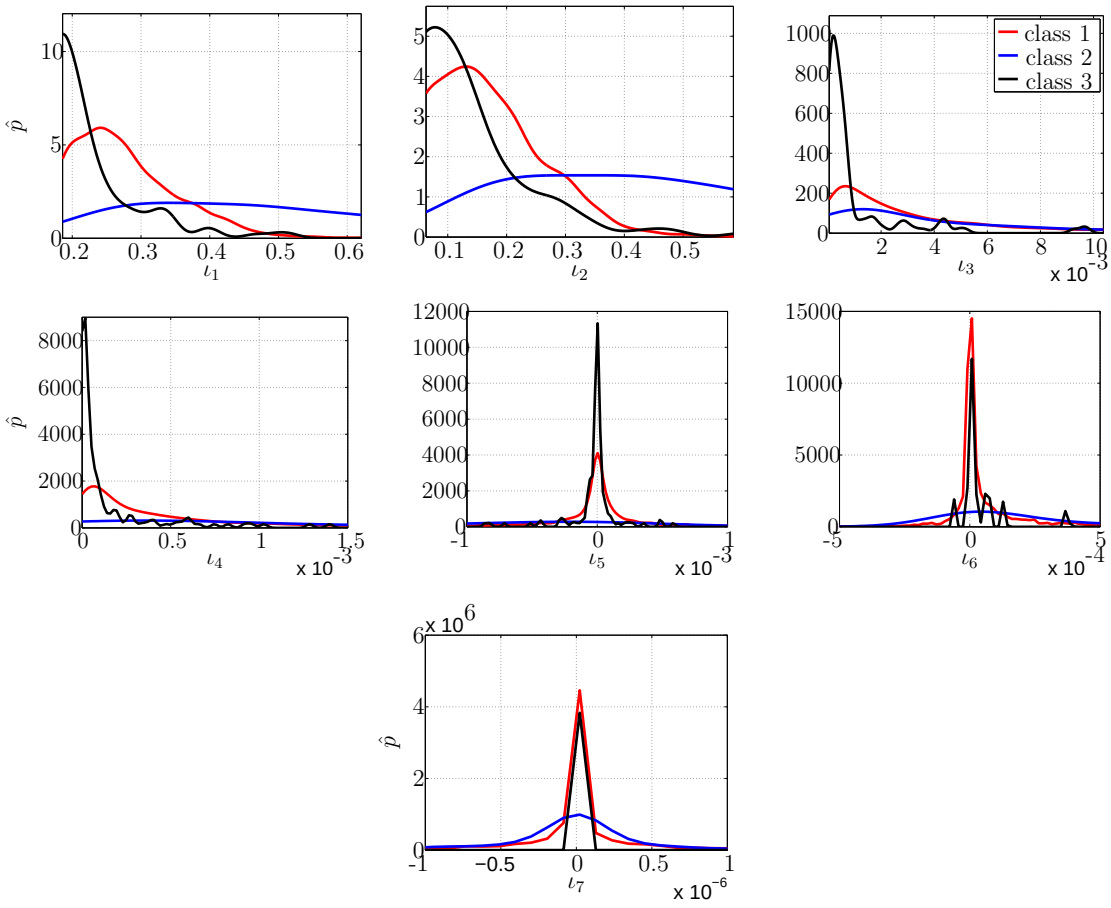


Figure A.5: Invariant moments

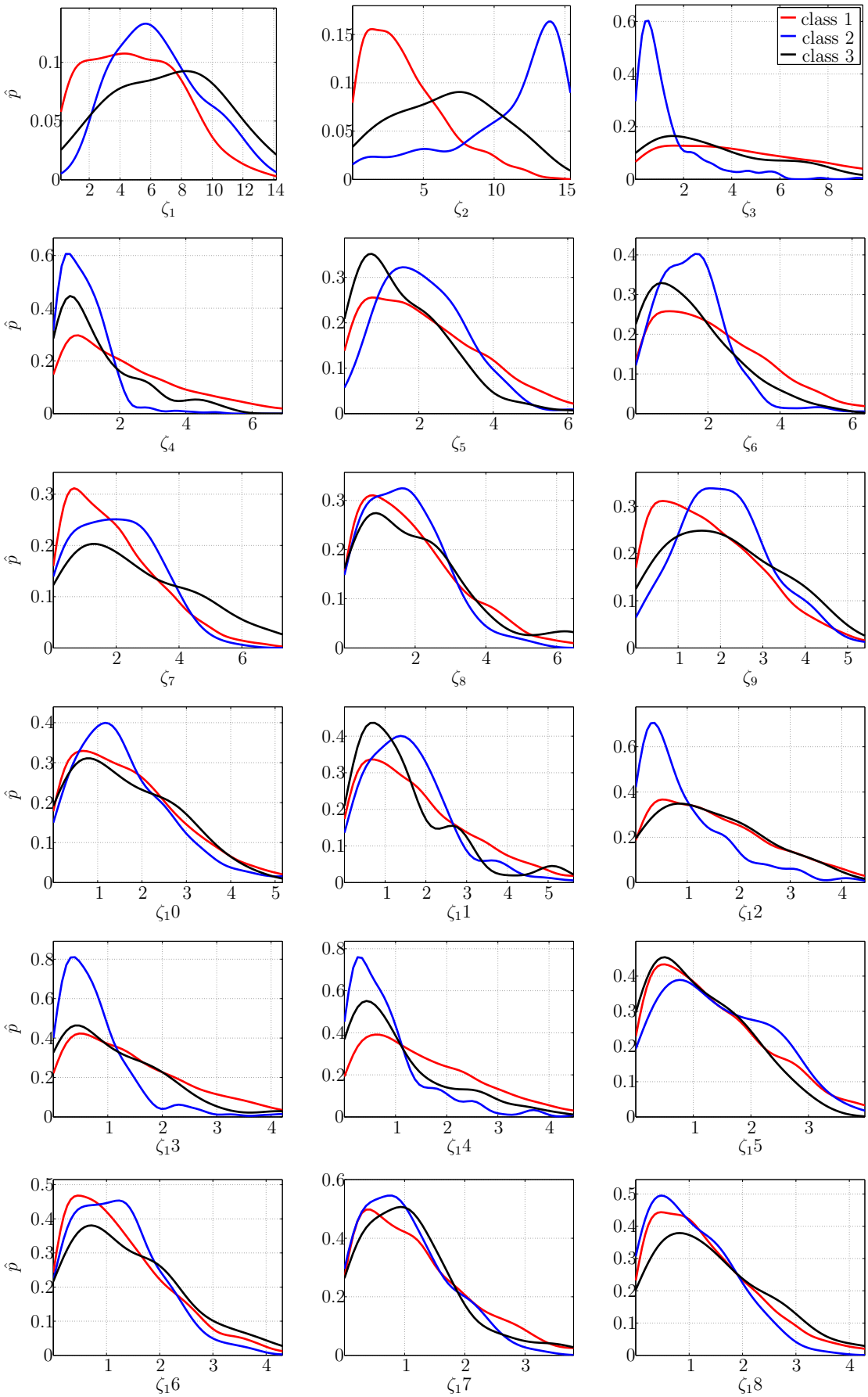


Figure A.6: Principal components

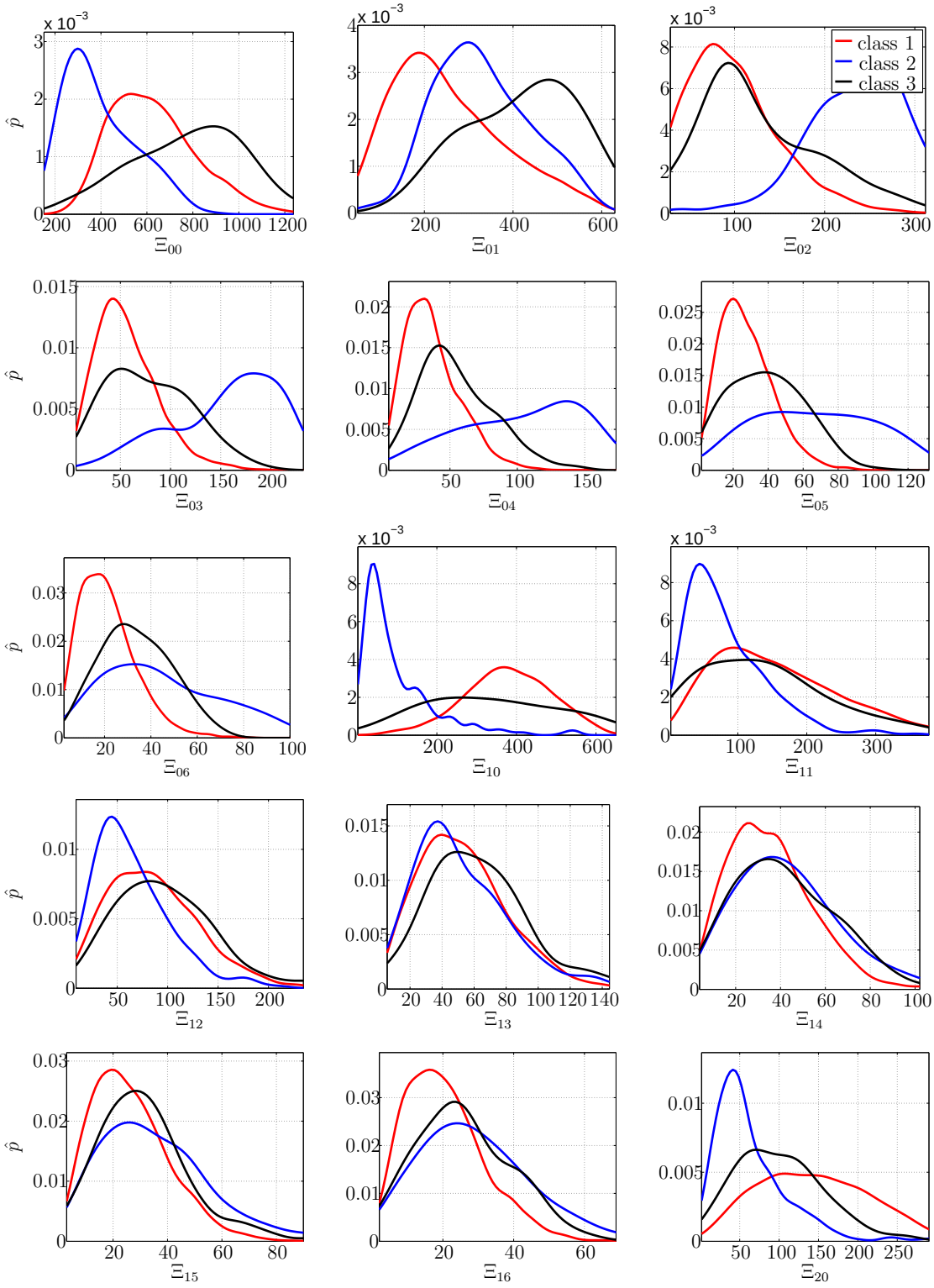


Figure A.7: 2D-Fourier coefficients

List of Acronyms

AC	Active Contours
ADAC	Automatic Detection And Classification
CAD/CAC	Computer Aided Detection and Classification
D-**SFS**	D Sequential Forward Selection
D-**SFFS**	D Sequential Forward Floating Selection
ICE	Iterative Conditional Estimation
ICM	Iterative Conditional Modes
KDE	Kernel Density Estimation
k-**NN**	k-Nearest Neighbours
LDA	Linear Discriminant Analysis
MAP	Maximum *A Posteriori*
MRF	Markov Random Fields
PCA	Principal Component Analysis
pdf	probability density function
SAS	Synthetic Aperture Sonar
SBS	Sequential Backward Selection
SFS	Sequential Forward Selection
SFFS	Sequential Forward Floating Selection
SNR	Signal-to-Noise Ratio
SVM	Support Vector Machines

List of Symbols

`bkg`	Background label (Chapter 7)
`hlt`	Highlight label (Chapter 7)
`sdw`	Shadow label (Chapter 7)
`tgt`	Target label (Chapter 7)
$\mathcal{B}$	Sink seeds (Chapter 7)
$\mathcal{E}$	Graph edges (Chapter 7)
$\mathcal{G}$	Graph (Chapter 7)
$\mathcal{L}$	Lattice (Chapter 7)
$\mathcal{M}$	Neighborhood (Chapter 7)
$\mathcal{O}$	Source seeds (Chapter 7)
$\mathcal{R}$	Rayleigh distribution
$\mathcal{S}$	Set of pixels labeled as `sdw` (Chapter 7)
$\mathcal{T}$	Set of pixels labeled as `bkg` (Chapter 7)
$\mathcal{V}$	Graph nodes (Chapter 7)
$\mathcal{W}$	Weibull distribution
a	Source pixel (Chapter 7)
$\mathbf{a}$	Set of active branches (Chapter 4)
b	Resampling index
$\mathbf{b}$	Contour of a region
c	Class variable
$\mathbf{c}$	Center of mass
f	Figure of merit
f^*	Figure of merit of the optimal feature subset $\mathbf{t}^*$
f_λ	Figure of merit considering class imbalance
$f_{\lambda,\lambda'}$	Figure of merit considering class imbalance and different mine classes
g	Weight/capacity of a graph edge (Chapter 7)
h	Parameter for graph cut initialization (Chapter 7)
h_B	Bandwith parameter of KDE
m	Sink pixel (Chapter 7)
$\mathbf{m}$	Vector for initialiazation (Chapter 7)
n	Number of elements in the feature subset

n^*	Number of elements in the optimal feature subset
o	Shadow orientation (Chapter 8)
p	pdf
$\hat{p}$	Estimated pdf
$p_{f\|n}$	pdf of f conditional on n
$p_{n\|f}$	pdf of n conditional on f
$p_{\mathbf{x}}$	pdf of label field $\mathbf{x}$
p_{x_i}	pdf of label of pixel i
$p_{\mathbf{x}\|\mathbf{y}}$	pdf of label field $\mathbf{x}$ conditional on image $\mathbf{y}$
$p_{\mathbf{y}}$	pdf of image $\mathbf{y}$
p_{y_i}	pdf of intensity of pixel i
$p_{\mathbf{y}\|\mathbf{x}}$	pdf of image $\mathbf{y}$ conditional on label field $\mathbf{x}$
$p_{\mathtt{bkg}}$	pdf of the intensity of background pixels
$p_{\mathtt{hlt}}$	pdf of the intensity of highlight pixels
$p_{\mathtt{sdw}}$	pdf of the intensity of shadow pixels
q	Quantile
$r_{\mathtt{hlt}}$	Ratio of principal axes of the highlight (Chapter 8)
$r_{\mathtt{sdw}}$	Ratio of principal axes of the shadow (Chapter 8)
$r_{\mathtt{sdw,hlt}}$	Ratio of shadow and highlight crossrange widths (Chapter 8)
r_η	Ratio of η_v and η_u (Chapter 8)
s	Observation index
t_j	Feature
$\mathbf{t}$	Feature set
$\mathbf{t}^*$	Optimal feature subset
u	Crossrange position
v	Range position
$w(f)$	Weighting function in Q
w_0	Hyperplane constant (Chapter 3)
$\mathbf{w}$	Hyperplane vector (Chapter 3)
x_i	Label of pixel i (Chapter 7)
$\mathbf{x}$	Label field in vector notation (Chapter 7)
y_i	Intensity of pixel i
$\mathbf{y}$	Sonar image in vector notation (Chapter 7)

A	Normalization constant (Chapter 3)

B	Boundary properties (Chapter 7)
C	Number of classes
D	Number of branches (Chapter 4)
E	Graph cost (Chapter 7)
F	Cost function (Chapter 7)
G	Gibbs energy (Chapter 7)
H	Histogram
$H_{f\|n}$	Histogram of f conditional on n
I	Ingoing edges (Chapter 7)
$\mathbf{I}$	Binary representation of the segmented shadow (Chapter 8)
J	Constant for rule decision of Mahalanobis' classifier
N	Number of elements in the feature set $\mathbf{t}$
N_{B}	Number of samples for resampling method (Chapter 3)
N_{C}	Rectangle length in crossrange direction (Chapter 7)
N_{Gibbs}	Number of Gibbs samples (Chapter 7)
N_{R}	Rectangle length in range direction (Chapter 7)
N_{sub}	Number of sub-images (Chapter 7)
N_{W}	Number of elements in $\mathbf{m}$ (Chapter 7)
O	Outgoing edges (Chapter 7)
P	Probability
P_{m}	Probability of misclassification
Q	Quality assessment of classifier performance
R	Regional properties (Chapter 7)
S	Number of observations
S_c	Number of observations of class c
$\mathbf{T}$	Matrix with D alternatives for $\mathbf{t}^*$ (Chapter 4)
U	Regional cost (Chapter 7)
U'	Boundary cost (Chapter 7)
V	Number of nodes of $\mathbf{b}$
V'	Number of nodes added per iteration (Chapter 7)
W	Graph cut (Chapter 7)
$\mathbf{X}$	Segmented sonar image
$\mathbf{Y}$	Sonar image
Z	Normalization constant (Chapter 7)

α	Parameter of the Rayleigh distribution
β_j	Parameter defining G (Chapter 7)
γ	Area of the shadow region (Chapter 8)
$\gamma_{\texttt{hlt}}$	Area of the highlight region (Chapter 8)
$\epsilon_{\texttt{hlt}}$	1/0 if the shadow has/has not a highlight (Chapter 8)
ζ_j	Principal component (Chapter 8)
μ	Mean
ν	Weighting factor (Chapter 7)
η	Neighbor of a pixel (Chapter 7)
η_u	Shadow width in range direction (Chapter 8)
η_v	Shadow width in crossrange direction (Chapter 8)
ι_j	Invariant moment (Chapter 8)
κ	SVM class constant
λ	Factor for the figure of merit f_λ
λ'	Factor for the figure of merit $f_{\lambda,\lambda'}$
ξ	Scale Weibull parameter
ξ'	Shape Weibull parameter
ρ	Perimeter (Chapter 8)
$\hat{\sigma}_{i,j}$	Sample central moment (Chapter 8)
$\bar{\sigma}_{i,j}$	Normalized central moment (Chapter 8)
τ	Segmentation overlap (Chapter 8)
υ	Lagrange multiplier
ϕ	Flow (Chapter 7)
χ	Compactness (Chapter 8)
ψ	Parameter of Q
ω_1	Boundary coef. 1 (Chapter 7)
ω_2	Boundary coef. 2 (Chapter 7)
Γ	Solidity (Chapter 8)
Δ_o	Orientation distance (Chapter 8)
$\Delta\mathbf{I}_v$	Cumulative projection of $\mathbf{I}$ on the crossrange direction (Chapter 8)
Θ_j	Paramter defining G (Chapter 7)
Λ	Topological skeleton
$\Xi_{i,j}$	2D-Fourier coefficient (Chapter 8)
$\Pi_{\texttt{hlt}}$	Length of the highlight (Chapter 8)

Υ	Hough transform
$\mathbf{\Sigma}$	Covariance matrix
Φ	Kernel function
Ψ	Geometrical feature (max $\{\Delta \mathbf{I}_v\}/\eta_v$)
$\mathbf{\Omega}_x$	Parameter vector of *a priori* probability (Chapter 7)
$\mathbf{\Omega}_y$	Parameter vector of likelihood function (Chapter 7)

Bibliography

[1] G. Magy, "State of the art in pattern recognition," in *Proceedings of IEEE*, 1968, vol. 56, pp. 836–862.

[2] L. Kanal, "Patterns in pattern recognition: 1968-1974," *IEEE Transactions on Information Theory*, vol. 20, no. 6, pp. 697–722, 1974.

[3] P. Devijver and J. Kittler, *Pattern Recognition: a Statistical Approach*, London: Prentice Hall, 1982.

[4] K. Fukunaga, *Introduction to Statistical Pattern Recognition (2nd ed.)*, Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[5] J. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*, 1992.

[6] L. Devroye, L. Gyorfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, 1996.

[7] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[8] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Computing Surveys (CSUR)*, vol. 35, no. 4, pp. 399–458, 2003.

[9] S. Z. Li and A. K. Jain, *Handbook of Face Recognition*, Springer-Verlag New York, Inc., 2011.

[10] A. K. Jain, S. Prabhakar, and L. Hong, "A multichannel approach to fingerprint classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 348–359, 1999.

[11] A. Zimek, F. Buchwald, E. Frank, and S. Kramer, "A study of hierarchical and flat classification of proteins," *Transactions on Computational Biology and Bioinformatics, IEEE/ACM*, vol. 7, no. 3, pp. 563–571, 2010.

[12] C. Debes, J. Hahn, A. M. Zoubir, and M. Amin, "Target discrimination and classification in through-the-wall radar imaging," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4664–4676, 2011.

[13] I. Quidu, J. Ph. Malkasse, G. Burel, and P. Vilbe, "Mine classification based on raw sonar data: an approach combining Fourier descriptors, statistical models, and genetic algorithms," in *Proceedings of the OCEANS Conference*, 2000, vol. 1, pp. 285–290.

[14] C. M. Ciany and W. Zurawski, "Performance of fusion algorithms for computer aided detection and classification of bottom mines in the shallow water environment," in *Proceedings of the OCEANS Conference*, 2002, vol. 4, pp. 2164–2167.

[15] S. Reed, Y. Petillot, and J. Bell, "Automated approach to classification of mine-like objects in sidescan sonar using highlight and shadow information," *IEE Proceedings - Radar, Sonar and Navigation*, vol. 151, no. 1, pp. 48–56, 2004.

[16] R. E. Bellman, "Adaptive control processes," *Princeton University Press*, 1961.

[17] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Transactions on Information Theory*, vol. 14, no. 1, pp. 55–63, 1968.

[18] T. M. Cover and J. M. Van Campenhout, "On the possible orderings in the measurement selection problem," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 7, no. 9, pp. 657–661, 1977.

[19] A. K. Jain, R. P. W. Duin, and M. Jianchang, "Statistical pattern recognition: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, 2000.

[20] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, pp. 491–502, 2005.

[21] Q.-L. Tran, K.-A. Toh, D. Srinivasan, K.-L. Wong, and S. Q.-C. Low, "An empirical comparison of nine pattern classifiers," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 35, no. 5, pp. 1079–1091, 2005.

[22] M. P. Sampat, A. C. Patel, Y. Wang, S. Gupta, C.-W. Kan, A. C. Bovik, and M. K. Markey, "Indexes for three-class classification performance assessment – an empirical comparison," *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 3, pp. 300–312, 2009.

[23] W. A. Yousef, R. F. Wagner, and M. H. Loew, "Comparison of non-parametric methods for assessing classifier performance in terms of ROC parameters," in *Proceedings of the Applied Imagery Pattern Recognition Workshop*, 2004, pp. 190–195.

[24] W. E. Weideman, M. T. Manry, H.-C. Yau, and W. Gong, "Comparisons of a neural network and a nearest-neighbor classifier via the numeric handprint recognition problem," *IEEE Transactions on Neural Networks*, vol. 6, no. 6, pp. 1524–1530, 1995.

[25] J. D. Paola and R. A. Schowengerdt, "A detailed comparison of backpropagation neural network and maximum-likelihood classifiers for urban land use classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 33, no. 4, pp. 981–996, 1995.

[26] A. K. Jain and B. Chandrasekaran, "Dimensionality and sample size considerations in pattern recognition practice," in *Classification Pattern Recognition and Reduction of Dimensionality*, P.R. Krishnaiah and L.N. Kanal, Eds., vol. 2 of *Handbook of Statistics*, pp. 835–855. Elsevier, 1982.

[27] B. Kim and D. A. Landgrebe, "Prediction of optimal number of features," in *Proceedings of the Annual International Geoscience and Remote Sensing Symposium*, May 1990, pp. 2393–2396.

[28] A. K. Jain and D. Zongker, "Feature selection: evaluation, application, and small sample performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 153–158, Feb. 1997.

[29] A. Frank and A. Asuncion, "UCI machine learning repository," 2010.

[30] B. D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge, 1996.

[31] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*, John Wiley and Sons, Inc., Hoboken, NJ, third edition, 2003.

[32] L. Breiman, J. H. Friedman, and R. A. Olshen, *Classification and Regression Trees*, Wadsworth, 1984.

[33] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[34] A. M. Zoubir and D. R. Iskander, *Bootstrap Techniques for Signal Processing*, Cambridge University Press, 2004.

[35] A. K. Jain, R. C. Dubes, and C. Chen, "Bootstrap techniques for error estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, pp. 628–633, 1987.

[36] S. Borra and A. Di Ciaccio, "Measuring the prediction error. a comparison of cross-validation, bootstrap and covariance penalty methods," *Computational Statistics & Data Analysis.*, vol. 54, pp. 2976–2989, 2010.

[37] H.-C. Chung and C.-P. Han, "Conditional confidence intervals for classification error rate," *Computational Statistics & Data Analysis*, vol. 53, no. 12, pp. 4358–4369, 2009.

[38] J. Dias and J.n Vermunt, "A bootstrap-based aggregate classifier for model-based clustering," *Computational Statistics*, vol. 23, no. 4, pp. 643–659, 2008.

[39] A. Isaksson, M. Wallman, H. Göransson, and M. G. Gustafsson, "Cross-validation and bootstrapping are unreliable in small sample classification," *Pattern Recognition Letters*, vol. 29, pp. 1960–1965, 2008.

[40] N. Diaz-Diaz, J. S. Aguilar-Ruiz, J. A. Nepomuceno, and J. Garcia, "Feature selection based on bootstrapping," in *Congress on Computational Intelligence Methods and Applications, ICSC*, 2005.

[41] M. Verleysen, F. Rossi, and D. François, "Similarity-based clustering," chapter Advances in Feature Selection with Mutual Information, pp. 52–69. Springer-Verlag, Berlin, Heidelberg, 2009.

[42] J.-L. Yuan, "Bootstrapping nonparametric feature selection algorithms for mining small data sets," in *International Joint Conference on Neural Networks*, 1999, vol. 4, pp. 2526–2529.

[43] J. P. Hoffbeck and D. A. Landgrebe, "Covariance matrix estimation and classification with limited training data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 763–767, 1996.

[44] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Annual Workshop on Computational Learning Theory*, 1992, pp. 144–152.

[45] B. Scholkopf, K. Tsuda, and J. P. Vert, *Kernel Methods in Computational Biology*, MIT Press, 2004.

[46] C. C. Chang and C. J. Lin, *LIBSVM: a library for support vector machines*, 2001.

[47] A. M. Zoubir and R. Iskander, "Bootstrap methods and applications," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 10–19, 2007.

[48] D. W. K. Andrews and M. Buchinsky, "A three-step method for choosing the number of bootstrap repetitions," *Econometrica*, vol. 68, no. 1, pp. 23–51, 2000.

[49] J. Shao and D. Tu, *The Jackknife and Bootstrap*, Springer-Verlag New York, Inc., 1995.

[50] D. N. Politis, J. P. Romano, and M. Wolf, *Subsampling (Springer Series in Statistics)*, Springer-Verlag New York, Inc., 1999.

[51] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, 1972.

[52] C. R. Rao, "On some problems arising out of discrimination with multiple characters," *Sankhya: The Indian Journal of Statistics (1933-1960)*, vol. 9, no. 4, pp. 343–366, 1949.

[53] T. L. Boullion, P. L. Odell, and B. S. Duran, "Estimating the probability of misclassification and variate selection," *Pattern Recognition*, vol. 7, no. 3, pp. 139–145, 1975.

[54] J. van Ness, "On the effects of dimension in discriminant analysis," *Technometrics*, vol. 18, pp. 175–187, 1979.

[55] A. K. Jain and R. Duber, "On the optimal number of features in the classification of multivariate Gaussian data," *Pattern Recognition*, vol. 10, pp. 365–374, 1978.

[56] S. Raudys and V. Pikelis, "On dimensionality, sample size, classification error, and complexity of classification algorithm in pattern recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-2, no. 3, pp. 242–252, 1980.

[57] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distributions," *Bulletin of the Calcutta Mathematical Society*, vol. 35, pp. 99–109, 1943.

[58] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998.

[59] C. Sima, S. Attoor, U. Brag-Neto, and E. R. Dougherty J. Lowey, E. Suh, "Impact of error estimation on feature selection," *Pattern Recognition*, vol. 38, no. 12, pp. 2472–2482, 2005.

[60] S. Kullback and R. A. Leibler, "On information and sufficiency," *Annals of Mathematical Statistics*, vol. 22, pp. 79–86, 1951.

[61] E. Parzen, "On estimation of a probability density function and mode," *Annals of Mathematical Statistics*, vol. 33, pp. 1065–1076, 1962.

[62] J. Reunanen, I. Guyon, and A. Elisseeff, "Overfitting in making comparisons between variable selection methods," *Journal of Machine Learning Research*, vol. 3, pp. 1371–1382, 2003.

[63] P. Somol, J. Novovicova, and P. Pudil, "Are better feature selection methods actually better? - discussion, reasoning and examples," in *HEALTHINF (1)*, 2008, pp. 246–253.

[64] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, pp. 1119–1125, 1994.

[65] A. W. Whitney, "A direct method of nonparametric measurement selection," *IEEE Transactions on Computers*, vol. 20, pp. 1100–1103, 1971.

[66] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[67] T. Marill and D. Green, "On the effectiveness of receptors in recognition systems," *IEEE Transactions on Information Theory*, vol. 9, no. 1, pp. 11–17, 1963.

[68] S. G. Johnson and A. Deaett, "The application of automated recognition techniques to side-scan sonar imagery," *IEEE Journal of Oceanic Engineering*, vol. 19, no. 1, pp. 138–144, 1994.

[69] M. P. Hayes, "Synthetic aperture sonar: A review of current status," *IEEE Journal of Oceanic Engineering*, vol. 34, no. 3, pp. 207–223, 2009.

[70] P. Blondel, *The handbook of sidescan sonar*, Springer Praxis Books, 2009.

[71] D. Massonnet and J. C. Souyris, *Imaging with Synthetic Aperture Radar*, CRC Press, 2008.

[72] C. A. Wiley, "Synthetic aperture radars: A paradigm for technology evolution," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-21, no. 3, pp. 440–443, 1985.

[73] D. P. Williams and E. Coiras, "On sand ripple detection in synthetic aperture sonar imagery," in *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2010, pp. 1074–1077.

[74] J. Chanussot, F. Maussang, and A. Hetet, "Scalar image processing filters for speckle reduction on synthetic aperture sonar images," in *Proceedings of the OCEANS Conference*, 2002, vol. 4, pp. 2294–2301.

[75] F. Langner, C. Knauer, W. Jans, and A. Ebert, "Side scan sonar image resolution and automatic object detection, classification and identification," in *Proceedings of the OCEANS Conference*, 2009, pp. 1–8.

[76] M. Mignotte, C. Collet, P. Pérez, and P. Bouthemy, "Sonar image segmentation using an unsupervised hierarchical MRF model," *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1216–1231, 2000.

[77] S. Reed, Y. Petillot, and J. Bell, "An automatic approach to the detection and extraction of mine features in sidescan sonar," *IEEE Journal of Oceanic Engineering*, vol. 28, no. 1, pp. 90–105, 2003.

[78] M. Mignotte, C. Collet, P. Pérez, and P. Bouthemy, "Hybrid genetic optimization and statistical model based approach for the classification of shadow shapes in sonar imagery," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 2, pp. 129–141, 2000.

[79] R. Balasubramanian and M. Stevenson, "Pattern recognition for underwater mine detection," in *Proceedings of the CAC/CAD Conference*, 2001.

[80] E. Dura, J. M. Bell, and D. M. Lane, "Superellipse fitting for the classification of mine-like shapes in side-scan sonar images," in *Proceedings of the OCEANS Conference*, 2002, vol. 1, pp. 23–28.

[81] J. Groen, E. Coiras, and D. Williams, "Detection rate statistics in Synthetic Aperture Sonar images," in *Proceedings of the Underwater Acoustic Measurements Conference*, 2009, pp. 367–374.

[82] E. Coiras and J. Groen, "3D target shape from SAS images based on a deformable mesh," in *Proceedings of the Underwater Acoustic Measurements Conference*, 2009, pp. 303–310.

[83] H. Midelfart, J. Groen, and O. Midtgaard, "Template matching methods for object classification in synthetic aperture sonar images," in *Proceedings of the Underwater Acoustic Measurements Conference*, 2009.

[84] E. Coiras, P.-Y. Mignotte, Y. Petillot, J. Bell, and K. Lebart, "Supervised target detection and classification by training on augmented reality data," *IET Radar Sonar Navigation*, vol. 1, no. 1, pp. 83–90, 2007.

[85] M. F. Doherty, J. G. Landowski, P. F. Maynard, G. T. Uber, D. W. Fries, and F. H. Maltz, "Side scan sonar object classification algorithms," in *Proceedings of the International Symposium on Unmanned Untethered Submersible Technology*, 1989, pp. 417–424.

[86] G. J. Dobeck, J. C. Hyland, and L. Smedley, "Automated detection and classification of sea mines in sonar imagery," in *Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 1997, vol. 3079, pp. 90–110.

[87] S. W. Perry and L. Guan, "Pulse-length-tolerant features and detectors for sector-scan sonar imagery," *IEEE Journal of Oceanic Engineering*, vol. 29, no. 1, pp. 138–156, 2004.

[88] T. Aridgides, M. F. Fernandez, and G. J. Dobeck, "Side-scan sonar imagery fusion for sea mine detection and classification in very shallow water," in *Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 2001, vol. 4394, pp. 1123–1134.

[89] K. Siantidis and U. Hölscher-Höbing, "A system for automatic detection and classification for a mine countermeasure AUV," in *Proceedings of the Underwater Acoustic Measurements Conference*, 2009.

[90] A. R. Castellano and B. C. Gray, "Autonomous interpretation of side scan sonar returns," in *Proceedings of the Symposium on Autonomous Underwater Vehicle Technology*, 1990, pp. 248–253.

[91] I. Quidu, J. P. Malkasse, G. Burel, and P. Vilbe, "Mine classification using a hybrid set of descriptors," in *Proceedings of the OCEANS Conference*, 2000, vol. 1, pp. 291–297.

[92] J. C. Delvigne, "Shadow classification using neural networks," in *Proceedings of the Undersea Defence Conference*, 1992, pp. 214–221.

[93] I. Tena Ruiz, D. Lane, and M. Chantler, "A comparison of inter-frame feature measures for robust object classification in sector scan sonar image sequences," *IEEE Journal of Oceanic Engineering*, vol. 24, pp. 458–469, 1999.

[94] F. Maussang, J. Chanussot, A. Hétet, and M. Amate, "Higher-order statistics for the detection of small objects in a noisy background application on sonar imaging," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 1, pp. 25–41, 2007.

[95] F. Maussang, M. Rombaut, J. Chanussot, and M. Amate, "Fusion of local statistical parameters for buried underwater mine detection in sonar imaging," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, 2008.

[96] G. J. Dobeck, "Algorithm fusion for automated sea mine detection and classification," in *Proceedings of the OCEANS Conference*, 2001, vol. 1, pp. 130–134.

[97] P.-Y. Mignotte, E. Coiras, H. Rohou, Y. Petillot, J. Bell, and K. Lebart, "Adaptive fusion framework based on augmented reality training," *IET Radar, Sonar and Navigation*, vol. 2, no. 2, pp. 146–154, 2008.

[98] B. Zerr, J. Fawcett, and D. Hopkin, "Adaptive algorithm for sea mine classification," in *Proceedings of the Underwater Acoustics Measurements Conference*, 2009, pp. 319–326.

[99] D. Williams and J. Groen, "Multi-view target classification in synthetic aperture sonar imagery," in *Proceedings of the Underwater Acoustics Measurements Conference*, 2009, pp. 699–704.

[100] J. Fawcett, V. Myers, D. Hopkin, A. Crawford, M. Couillard, and B. Zerr, "Multi-aspect classification of sidescan sonar images: Four different approaches to fusing single-aspect information," *IEEE Journal of Oceanic Engineering*, vol. 35, no. 4, pp. 863–876, 2010.

[101] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004.

[102] M. Pinto and A. Bellettini, "Shallow water synthetic aperture sonar: an enabling technology for NATO MCM forces," in *Proceedings of the Undersea Defence Technology Conference*, 2007.

[103] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[104] S. Guillaudeux, S. Daniel, and E. Maillard, "Optimization of a sonar image processing chain: a fuzzy rules based expert system approach," in *Proceedings of the OCEANS Conference*, 1996, vol. 3, pp. 1319–1323.

[105] S. Daniel, S. Guillaudeux, and E. Maillard, "Adaptation of a partial shape recognition approach," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics: 'Computational Cybernetics and Simulation'*, 1997, vol. 3, pp. 2157–2162.

[106] J. Besag, "On the statistical analysis of dirty images," *Journal of the Royal Statistical Society*, vol. B-48, pp. 259–302, 1986.

[107] S. Dugelay, C. Graffigne, and J. M. Augustin, "Deep seafloor characterization with multibeam echosounders by image segmentation using angular acoustic variations," in *Proceedings of the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 1996, vol. 2823, pp. 255–266.

[108] V. Murino, A. Trucco, and C. S. Regazzoni, "A probabilistic approach to the coupled reconstruction and restoration of underwater acoustic images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 9–22, 1998.

[109] V. Murino and A. Trucco, "Edge/region-based segmentation and reconstruction of underwater acoustic images by markov random fields," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1998, pp. 408–413.

[110] C. Collet, P. Thourel, P. Pérez, and P. Bouthemy, "Hierarchical MRF modeling for sonar picture segmentation," in *Proceedings of the International Conference on Image Processing*, 1996, vol. 3, pp. 979–982.

[111] P. Thourel, C. Collet, P. Bouthemy, and P. Pérez, "Multiresolution analysis and MRF modeling applied to the segmentation of shadows in sonar pictures," in *Proceedings of the Asian Conference in Computer Vision*, 1996, vol. 2, pp. 81–85.

[112] F. Salzenstein and W. Pieczynski, "Parameter estimation in hidden fuzzy Markov Random Fields and image segmentation," *Graphical Models Image Processing*, vol. 59, no. 4, pp. 205–220, 1997.

[113] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, pp. 321–331, 1988.

[114] M. Lianantonakis and Y. R. Petillot, "Sidescan sonar segmentation using active contours and level set methods," in *Proceedings of the OCEANS Conference*, 2005, vol. 1, pp. 719–724.

[115] K. Imen, R. Fabler, J.-M. Boucher, and J.-M. Augustin, "Region-basedand incidence angle dependent segmentation of seabed sonar images using a level set approach combined to local texture statistics," in *Proceedings of the OCEANS - Asia Pacific Conference*, 2006, pp. 1–7.

[116] J. Besag, "Spatial interaction and statistical analysis of Lattice systems," *Journal of the Royal Statistical Society*, vol. 36, pp. 192–236, 1974.

[117] F. Spitzer, "Markov random fields and Gibbs ensembles," *The American Mathematical Monthly*, vol. 78, no. 2, pp. 142–154, 1971.

[118] H. Derin and H. Elliott, "Modeling and segmentation of noisy and textured images using Gibbs random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 1, pp. 39–55, 1987.

[119] M. Mignotte, C. Collet, P. Pérez, and P. Bouthemy, "Three-class Markovian segmentation of high resolution sonar images," *Computer Vision and Image Understanding*, vol. 76, no. 3, pp. 191–204, 1999.

[120] W. I. Stewart, D. Chu, S. Malik, S. Lerner, and H. Singh, "Quantitative seafloor characterization using a bathymetric sidescan sonar," *IEEE Journal of Oceanic Engineering*, vol. 19, no. 4, pp. 599–610, 1994.

[121] F. Maussang, J. Chanussot, A. Hetet, and M. Amate, "Mean / standard deviation representation of sonar images for echo detection: Application to SAS images," *IEEE Journal of Oceanic Engineering*, vol. 32, no. 4, pp. 956–970, 2007.

[122] A. C. Cohen, "Maximum likelihood estimation in the Weibull distribution based on complete and on censored samples," *Technometrics*, vol. 7, no. 4, pp. 579–588, 1965.

[123] Y. Boykov, O. Veksler, and R. Zabih, "Markov Random Fields with efficient approximations," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 1998, pp. 648–655.

[124] L. R. Ford and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, 1962.

[125] Y. Boykov, O. Veksler, and R. Zybih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 1222–1239, 2001.

[126] T. F. Cootes, A. Hill, C. J. Taylor, and J. Haslam, "Use of active shape models for locating structures in medical images," *Image and Vision Computing*, vol. 12, no. 6, pp. 355–365, 1994.

[127] Y. Amit, U. Grenander, and M. Piccioni, "Structural image restoration through deformable templates," *Journal of the American Statistical Association*, vol. 86, no. 414, pp. 376–387, 1991.

[128] C. Kervrann and F. Heitz, "A hierarchical statistical framework for the segmentation of deformable objects in image sequences," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1994, pp. 724–728.

[129] C. Chesnaud, P. Refregier, and V. Boulet, "Statistical region snake-based segmentation adapted to different physical noise models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1145–1157, 1999.

[130] MATLAB, *version 7.7.0.471 (R2008b)*, The MathWorks Inc., Natick, Massachusetts, 2008.

[131] J. Fawcett, "Image-based classification of sidescan sonar detections," in *Proceedings of the CAC/CAD Conference*, 2001.

[132] L. Da Fontoura Costa and R. Marcondes Cesar, *Shape Analysis and Classification: Theory and Practice (Image Processing Series)*, CRC Press, Taylor & Francis Group, 6000 Broken Sound Parkway NW, Suite 300, 2000.

[133] D. Zhang and G. Lu, "Review of shape representation and description techniques," *Pattern Recognition*, vol. 37, no. 1, pp. 1–19, 2004.

[134] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[135] M. Timothy, *Signal and Image Processing with Neural Networks*, John Wiley & Sons, Inc., New York, NY, USA, 1994.

[136] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.

[137] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, no. 6, pp. 559–572, 1901.

[138] P. M. Mather and T. Brandt, *Classification Methods for Remotely Sensed Data, Second Edition*, CRC Press, Taylor & Francis Group, 6000 Broken Sound Parkway NW, Suite 300, 2009.

[139] D. Boulinguez and A. Quinquis, "Classification of underwater objects using Fourier descriptors," in *Proceedings of the International Conference on Image Processing and its Applications*, 1999, pp. 240–244.

Curriculum vitae

Name:	Raquel Fandos
Date of birth:	26.05.1979
Place of birth:	Zaragoza (Spain)
Family status:	single

Education

10/1997-09/2003	Universidad de Zaragoza (Zaragoza, Spain) Telecommunications Engineering (Master of Science)
10/2002-07/2003	Kungliga Tekniska Högskolan (Stockholm, Sweden) Erasmus Grant, Master Thesis
06/2000	High school degree (Abitur) at Instituto Miguel Servet, Zaragoza, Spain

Work experience

04/2009 - 03/2012	Research associate at Signal Processing Group Technische Universität Darmstadt
01/2005 - 06/2008	Research and development RF engineer at CERN for the feasibility study of the linear accelerator CLIC in Geneva, Switzerland
04/2004 - 12/2004	Employed at Accenture as a system analyst in the field of ERP (Execution Resource Planning) in Barcelona, Spain

Erklärung laut §9 der Promotionsordnung

Ich versichere hiermit, dass ich die vorliegende Dissertation allein und nur unter Verwendung der angegebenen Literatur verfasst habe. Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

Darmstadt, 5. Dezember 2011,